HIGH SPEED DATA ACQUISITION SYSTEM FOR COMPUTER TOMOGRAPHIC IMAGING INSTRUMENTATION

VOT 74188

Professor Dr. Ruzairi Bin Hj. Abdul Rahim
Faculty of Electrical Engineering
Universiti Teknologi Malaysia

DECEMBER 2005

ACKNOWLEDGEMENT

In preparing this report, many individuals and authorities have offered much help and support to make this research a success. I would like to take the opportunity here to express my greatest gratitude to all of them.

First of all, I would like to thank the Malaysian Government & Universiti Teknologi Malaysia especially Research & Management Centre UTM is highly appreciated and without which this research could not have been possible

Other individuals whom I would like to thank are Mr. Chiam Kok Thiam, Mr Chan Kok San and Mr Pang Jon Fea for their opinion and suggestions in handling problems faced during this research. Not to be forgotten are PROTOM Researchers, Ms Leong Lai Chen, Ms Goh Chiew Loon, Mr Tee Zhen Cong, Mr Mohd Hafiz Fazalul Rahiman and Mr Ng Wei Nyap.

. Thank you also to other individuals who have helped me.

ABSTRACT

Monitoring solid-gas flow in terms of flow visualization and mass flow rate (MFR) measurement is essential in industrial processes. Optical tomography provides a method to view the cross sectional image of flowing materials in a pipeline conveyor. Important flow information such as flow concentration profile, flow velocity and mass flow rate can be obtained without the need to invade the process vessel. The utilization of powerful computer together with expensive Data Acquisition System (DAQ) as the processing device in optical tomography systems has always been a norm. However, the advancements in silicon fabrication technology nowadays allow the fabrication of powerful Digital Signal Processors (DSP) at reasonable cost. This allows the technology to be applied in optical tomography system to reduce or even eliminate the need of personal computer and the DAQ. The DSP system was customized to control the data acquisition of 16x16 optical sensors (arranged in orthogonal projection) and 23x23 optical sensors (arranged in rectilinear projections) in 2 layers (upstream and downstream). The data collected was used to reconstruct the cross sectional image of flowing materials inside the pipeline, velocity profile measurement and mass flow rate measurement. The mass flow rate result is sent to a Liquid Crystal Display (LCD) display unit for display. For image display purpose, the reconstructed image was sent to a personal computer via serial link. This allows the use of a laptop to display the cross sectional image besides performing any other tasks such as data logging.

ABSTRAK

Pengawasan aliran pepejal-gas dari segi gambaran aliran dan pengukuran kadar aliran jisim adalah penting dalam proses industri. Tomografi optik menyediakan satu kaedah untuk melihat gambaran keratan rentas aliran bahan dalam paip. Maklumat penting aliran seperti profil kepadatan, kelajuan aliran dan kadar aliran jisim boleh diperolehi tanpa perlu mencerobohi paip proses. Penggunaan komputer berkuasa tinggi bersama-sama dengan Sistem Pemerolehan Data yang mahal sebagai perkakasan pemproses dalam tomografi optik sudah menjadi kebiasaan. Walau bagaimanapun, kecanggihan dalam teknologi penghasilan silikon kini membolehkan penghasilan Pemproses Isyarat Digital yang berkuasa tinggi pada kos yang berpatutan. Ini membolehkan teknologi ini diaplikasikan dalam tomografi optik bagi mengurangkan atau mengelakkan sama sekali pergantungan pada komputer dan Sistem Pemerolehan Data. Sistem Pemproses Isyarat Digital dikonfigurasikan untuk mengawal proses pemerolehan data bagi 16x16 sensor optik (disusun dalam projeksi 'orthogonal') dan 23x23 sensor optik (disusun dalam projeksi 'rectilinear') pada 2 lapisan (atas dan bawah). Data yang dikumpul digunakan untuk menghasilkan gambaran keratan rentas aliran bahan dalam paip, pengukuran profil kelajuan dan pengukuran kadar aliran jisim. Keputusan kadar aliran jisim dihantar ke unit pemapar untuk tujuan paparan. Untuk tujuan gambaran imej, imej yang dihasilkan dihantar ke komputer melalui talian sesiri. Ini membolehkan penggunaan komputer riba untuk memaparkan imej keratan rentas selain menjalankan tugas lain seperti merekodkan data.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS / ABBREVIATIONS

ADC - Analogue to Digital Converter
ALU - Arithmetic Logic Unit
ASIC - Application Specific Integrated Circuit
bps - Bit Per Second
CAN - Controller Area Network
CODEC - Coder Decoder
CPU - Central Processing Unit
DAC - Digital to Analogue Converter
DFT - Discrete Fourier Transform
DIF - Decimation In Frequency
DIT - Decimation In Time
DSP - Digital Signal Processor
ECT - Electrical Capacitance Tomography
EEPROM - Electrical Erasable Programmable Read Only Memory
EIT - Electrical Impedence Tomography
EPROM - Electrical Programmable Read Only Memory
FFT - Fast Fourier Transform
FPGA - Field Programmable Gate Array
FPU - Floating Point Unit
GPIO - General Purpose Input Output
GUI - Graphical User Interface
HC11 - M68HC11 micro controller
HPI - Host Processor Interface
Hz - Hertz
IFFT - Inverse Fourier Transform
ISR - Interrupt Service Routine
LCD - Liquid Crystal Display

LED - Light Emitting Diode
LSB - Least Significant Bit
MAC - Multiply and Accumulate
MFR - Mass Flow Rate
MIPS - Million Instructions Per Second
MRI - Magnetic Resonance Imaging
ms - mili seconds
MSB - Most Significant Bit
MSPS - Mega Samples Per Second
NMR - Nuclear Magnetic Resonance
PC - Personal Computer
PCB - Printed Circuit Board
PSoC - Programmable System on Chip
PT - Process Tomography
PTP - Pixel To Pixel
RAM - Random Access Memory
RFFT - Fast Fourier Transform for Pure Real Signals
SAC - Succesive Approximation Conversion
SCI - Serial Communication Interface
SPI - Serial Peripheral Interface
SRAM - Static Random Access Memory
STS - Sensor To Sensor
us - micro seconds
USART - Universal Serial Asynchronous Receiver Transmitter
USB - Universal Serial Bus

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Process Tomography Overview

The term tomography originated from two Greek words which were '*tomos*' and '*graph*'. '*Tomos*' means slice while '*graph*' means picture. Thus, the term tomography means a slice of a picture (Rahim and Green, 1997). In other words, tomography can be defined as an approach to visualize the cross sectional image of an object (Pang, 2004a). The concept of tomography was first applied in medical field to diagnose lung disorders in the early 1930s (Dickin *et al.*, 1992). Two commonly known tomography system used for medical diagnosis are X-ray tomography and the Magnetic Resonance Imaging (MRI).

During year 1980's, researches from University of United Manchester Institute of Science and Technology (UMIST) began to apply the concept of tomography into industrial process monitoring system; where the investigated objects were the pipeline conveyors and process vessels. This field of application is commonly known as Industrial Process Tomography (IPT) or Process Tomography (PT) (Pang, 2004a). Important flow information such as flow concentration, flow velocity and mass flow rate can be determined without the need to invade process or object. This information helped industries in monitoring, verifying and improving their pipeline conveyor and process vessels operation (Leong, 2004). In this research, knowledge in process tomography had been applied as the main concept to perform mass flow rate measurement.

1.2 Background Problems

The full automation of solid-handling plant has frequently not been possible because of the lack of a basic flow meter. Previously used technique to measure solid flow often involved removing the material for weighing which was complicated, expensive and did not allow continuous feedback for automatic control (Green *et al.,* 1978). Many non intrusive techniques attempted to measure solids mass flow in pneumatic conveyor were elaborated Beck *et al*. (1987).

Later, Xie *et al.* (1989) made an effort to measure mass flow of solids in pneumatic conveyor by combining electrodynamics and capacitance transducers. Rahim *et al.* (1995) investigated applications of fibre optics into optical tomography for flow concentration measurement. Chan (2002) followed up and investigated mass flow rate measurement using optical tomography by relating the flow's concentration profile measurement with its mass flow rate. This method first performs calibration by recording the flow's concentration percentage plotted against mass flow rate measured using weight scale and stop watch. Based on their findings, the two variables had linear relation in light flow condition. However, this technique requires re-calibration each time a new flow material is used.

Cross correlation technique was proposed by Beck *et al.* (1987) for flow velocity measurement. This function was successfully applied by Ibrahim and Green (2002a) for offline bubble flow velocity measurement using optical tomography. In 2003, Pang (2004a) implemented real time system for solid-gas flow concentration, flow velocity and mass flow rate measurement using optical tomography and data distribution system. The cross correlation techniques being used by these researches were based on time domain cross correlation function, which was very time consuming.

Pang (2004a) implemented four projections arrays (2 orthogonals and 2 rectilinears) in optical tomography system. The extra two layers (the Rectilinear layers) were used to filter the ambiguous image and smearing effects. However, analogue acquisition was used to acquire all the measurements. The obtained data

from masking layer were used for comparison to predetermined threshold values. Using this method, valuable time was utilized for analogue to digital conversions. Furthermore, the number of sensors on the rectilinear was quite large. Although the research successfully implemented real time optical tomography system, it required four powerful personal computers and a network hub in order to implement data distribution system. The resulting system was large and not easily portable.

1.3 Problem Statements

i. Data acquisition and analysing data in optical tomography requires fast digitizer and high performance computing power. In order to eliminate the use of expensive Data Acquisition System for sensor measurements and a powerful computer for data analysis, a high performance embedded system need to be designed and implemented. This embedded system must be able to perform data acquisition, analyse the measured data and support standard communication protocols. It is also the main controller for the overall operation.

ii. Signal measurement using Analogue to Digital Converter requires very long time, especially when the number of sensors is large. Alternative method such as voltage comparator to digitally sample signals on masking layer need to be investigated and applied. A method to send the measurements quickly to the embedded system also needs to be designed and implemented.

iii. Algorithms in processing acquired measurement data such as Pixel to Pixel correlation to obtain flow velocity is very computation intensive especially in this research where two layers of sensors totalling up to 156 sensors are used. Performing cross correlation to obtain velocity profile with only one hundred sets of data already requires thousands of calculations. New techniques and algorithms need to be investigated to reduce the processing time required in order to achieve a high performance system.

iv. The implementation of this research without using a computer as the main controller hinders user from changing settings or viewing results

acquired from the embedded system. Therefore, a user interface unit need to be designed and implemented for user input and output processing. Besides that, a tomography system without flow visualization seems odd. Therefore, a GUI need to be developed to serve the purpose of displaying information for debugging and flow monitoring.

v. Programming techniques affects the performance of an embedded system significantly. Deep understanding of memory architecture in the embedded system and code execution is required to develop optimized firmware for the system.

1.4 Importance of Study

The automation of solid material handling may be realized in the near future by having an instrument that is able to provide online solid-gas mass flow measurement. Online measurement eliminates the need for removing material for weighing which is complicated and expensive. Previous method of mass flow rate measurement using inverse solution of concentration measurement (Chan, 2002) requires calibration each time a new material is used or there is modification to the flow rig. This research aims to implement algorithm based mass flow rate measurement which will not require any calibration constant.

The implementation of this research using embedded systems results in a solid-gas mass flow rate meter that is small and portable. The overall cost will also be significantly reduced compared to using standard Data Acquisition System and powerful computers (Pang, 2004a). For industrial applications that are only interested in the mass flow rate measurement, it can be seen on a LCD. On the other hand, the instrument is also capable to visualize the material flow by connecting the instrument to a laptop via standard serial communication interface (RS 232). The Graphical User Interface installed on the laptop will display the current reconstructed image, mass flow rate measurement or sensor values for data logging or monitoring purpose.

Velocity profile was previously obtained using cross correlation in work by Xie *et al.* (1989), Ibrahim and Green (2002a) and Pang (2004a). Pixel to pixel correlation and sensor to sensor correlation were investigated by Ibrahim using offline system. Later, Pang implemented online system with real time measurement. Both Ibrahim and Pang showed that the periphery velocity could be obtained by cross correlating the upstream sensor layer with the downstream sensor layer. The fact that sensor to sensor correlation is much faster than pixel to pixel correlation motivates investigation of using the periphery velocity to obtain the velocity profile. By using this technique, significant amount of processing time can be reduced and therefore the overall system performance can be increased too.

The cross correlation technique used in this research is optimized by performing the calculations in frequency domain (known as Fast Convolution). Commonly, cross correlation is performed in time domain as it is less complicated and resembles a direct implementation of the cross correlation equation. By performing Fast Convolution, the processing time required to obtain velocity result reduces considerably as the number of cross correlation elements becomes larger. Noting that the input sequences for the cross correlation function are pure real values (not complex values) from the sensor data, the cross correlation function is optimized further using Fast Fourier Transform for pure real sequences (RFFT). This again reduces the processing time required to convert the data to frequency domain. In short, the results obtained from these investigations of using sensor to sensor cross correlation technique together with performing optimized cross correlation in frequency domain will significantly reduce the time required per mass flow rate measurement.

Last but not least, the development of the firmware for the embedded systems was optimized to achieve highest speed possible. Function calls, algorithms implementations and communication between sub systems were all developed with considerations for accuracy and optimum performance. These optimizations are very important as they affect the overall performance considerably.

1.5 Research Objectives

This research aimed to develop an embedded system based solid-gas mass flow rate meter using optical tomography technique. This research was carried out according to the following objectives:

i. To implement embedded system as the main controller in an optical tomography system for the purpose of data acquisition, data analysis and result display.

ii. To increase data acquisition rate by using high speed ADC for analogue sensor measurements and voltage comparators for sensors on masking layer.

iii. To optimize the performance of hybrid image reconstruction algorithm in terms of accuracy and processing speed when carried out on embedded system.

iv. To investigate the application and performance of cross correlation function carried out in frequency domain to obtain flow velocity.

v. To investigate the application and performance of sensor to sensor cross correlation results in reconstructing flow velocity profile.

vi. To implement algorithm based mass flow rate measurement using embedded system.

vii. To investigate the performance of the overall system in carrying out image reconstruction, velocity profile measurement and mass flow rate measurements.

viii. Finally, restudy the research result and make necessary suggestions and recommendations for further improvement.

1.6 Research Scopes

The scope of this research can be categorized as follows:

i. Design and implementation of an embedded system with capability to process instructions at high speed, perform digital inputs and outputs, support standard communication protocols (SPI, SCI) and contain sufficient memory to store program and large amount of measurement data.

ii. Design and implementation of signal conditioning circuit for digital sampling and analogue sampling. Digital sampling is used for masking purpose while the analogue sampling is needed to reconstruct the cross sectional image of the flowing material. The Signal Conditioning Unit must be able to multiplex 64 analogue signals to ADC channels and 92 binary data to SPI channel on the embedded system.

iii. Design and implementation of a User Interface Unit. This unit decodes keypad press and drives a 2-line, 16-character Liquid Crystal Display (LCD) panel to show the mass flow rate result and system settings. This unit should support RS 232 communication with the main controller and perform user input and output with minimum dependency to the main controller.

iv. Development of PC Graphical User Interface (GUI) software to be executed on a host computer. This GUI should serve the purpose of displaying information for debugging such as sensor values and current settings for the embedded system. Flow monitoring should also be supported by enabling the display of current reconstructed image and mass flow rate result. The interface between the GUI and the embedded system is based on standard RS 232 serial interface.

v. Analysis and investigation on performance of algorithms and implementations of velocity profile construction using sensor to sensor correlation, pixel to pixel correlation, processing in time domain and processing in frequency domain.

vi. Optimization on the overall system in carrying out image reconstruction, velocity profile measurement and mass flow rate measurement using programming techniques.

vii. Verification of results obtained by comparing between experiment results to theoretical calculations.

1.7 Report Organization

Chapter 1 provides an introduction to process tomography, research objectives, background problems, scope of study and the importance of study.

Chapter 2 reviews common types of process tomography systems, brief history of process tomography based mass flow rate measurement, flow velocity measurement techniques and embedded system technology.

Chapter 3 focuses on the hardware implementation of this research. The sensor configuration, signal conditioning circuits and architecture of the embedded systems are described.

Chapter 4 describes the software and firmware development to support the developed system. The data acquisition process, image reconstruction algorithm, cross correlation function, velocity profile measurement, mass flow rate measurement and communications between modules are explained thoroughly.

Chapter 5 presents the results obtained from experiments in terms of accuracy and processing speed. These results comprise of image reconstruction results, velocity measurement result and mass flow rate measurement results.

Chapter 6 states the conclusions made from this research and recommendations for future research.

CHAPTER 2

AN OVERVIEW OF PROCESS TOMOGRAPHY

Process Tomography provides real time cross sectional images of the distribution of materials in process. By analysing two suitably spaced images, it will be possible to measure the direction and speed of material movement (Dickins *et* al., 1992). Basically, a typical tomography system consists of sensor array, data acquisition system, image reconstruction and display system (Ibrahim *et al.*, 2000) as shown in Figure 2.1. For different types of flow conditions, different sensors and processing techniques are more suitable. This chapter aims to introduce common technologies being used in process tomography.

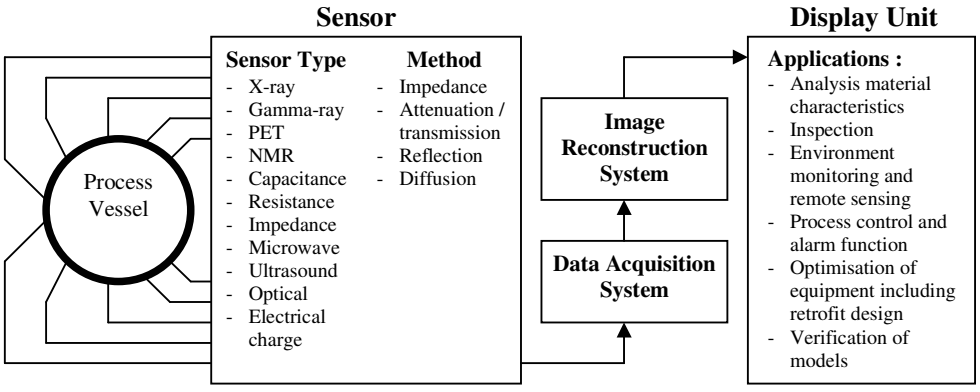


Figure 2.1 : Block Diagram of Typical Tomography System

2.1 Introduction

The difference between many process tomography systems is mainly the type of sensor and therefore the processing system being used. The selection of sensor type is usually determined by considering the properties or characteristic of the flow material being investigated such as electrical conductivity and flow state (liquid, gas or solid). A brief discussion of many types of tomography sensors were presented by Rahim and Green (1997). On the other hand, the selection of the processing system depends on the design requirements, such as cost, the need for high speed processing, the accuracy of digitizer and the available resources for overall system development. During processing, types of algorithms implemented depend on the required flow information such as concentration profile, flow velocity or mass flow rate. Dickins *et al.* (1992) discussed in details many applications and constraints of general tomography system.

2.2 Types of Tomography Systems

Sensor technology is the most essential and critical part of any measurement system (Rahim and Green, 1997). Basically, the sensor selection depends on the properties to be measured. For example, capacitance sensor is suitable for electrically non conducting fluid flow while impedance sensor requires that the flow material to be electrically conducting. There is also case where solids transfer in pneumatic systems develops static charge which can be detected using charge sensor (Yan, 1996). Six common tomography systems are described in the following.

2.2.1 Electrical Impedance Tomography (EIT)

Electrical Impedance Tomography (EIT) was originally developed for clinical applications, particularly imaging of lung functions (Azrita, 2002). In EIT systems, electrodes are placed into a vessel wall in order to make measurements of the distribution of electrical resistance within an object plane (Rahim and Green, 1997). Small current is injected through sensors encompassing the area to be

imaged. The resulting voltage drops are then measured on adjacent pairs of the remaining electrodes (Pang, 2004a). From the electrical point of view, the EIT measurements is the classic four-wire technique of resistance measurement (Zbigniew Rucki *et al.*, 2000).

The main advantage of EIT systems is its low cost compared to other techniques such as computerized X-ray or Positron Emission Tomography. On the other hand, the major disadvantage of EIT systems is its low spatial resolution. Among applications being developed using EIT concepts are imaging the distribution of oil/water or gas/oil in a pipeline, imaging of substance flow in a mixing vessel, mixing of meat chunks in pet food and mixing of fruit pips in jam Pang (2004a).

2.2.2 Electrical Capacitance Tomography (ECT)

An ECT system basically consists of three sub-systems, the primary capacitance sensor with multiple electrodes, the capacitance data acquisition electronics and the computer system for image reconstruction and process data interpretation (Rahim and Green, 1997).

The capacitance electrodes can be mounted either inside or outside the vessel, depending on the material of the vessel. Electrodes are normally mounted on the outside surface of non electrical conducting material pipe or vessel. This method offers non-invasive measurement and can be made intrinsically safe. The sensor is usually surrounded by an earthed shield to minimize the influence of external electrical fields or noise.

The presence of flow materials inside the capacitive sensing volume will change the measured capacitance (Yan, 1996). This change in the capacitance can be converted into a suitable electrical signal (current, voltage or frequency) by incorporating the ‘capacitor’ within a suitable electronic circuit such as an AC deflection bridge.

The major advantage of a capacitive sensor is its low cost, non invasive and safe. On the other hand, ECT system has the disadvantage of low resolution and is not suitable to be used in conductive materials (Leong, 2004).

2.2.3 Electrical Charge Tomography

Electrical charge tomography is also known as electrodynamic tomography (Mohd Hezri, 2002). Electrodinamics can be defined as the effects of moving charges as opposed to electrostatics whereby the charges are stationary (Leong, 2004).

The working principle in this type of tomography system is that the flow materials will be electrostatically charged during the transportation, primarily by virtue of friction of fine particles amongst themselves and abrasion on the walls of the conveyor. The electrical charge sensors which are mounted in the wall of the conveyor detect the induced charge and convert them to voltage readings (Green *et al*, 1997). These signals are amplified and transferred to data acquisition system to the computer for image reconstruction.

Electrical Charge Tomography has the advantage of being robust, low cost and sensitive to low flow rates of dry solid materials (Green *et al*, 1997). On the other hand, electrodynamics sensor is very sensitive to environmental noise and limited to applications in pneumatic systems only.

2.2.4 Ultrasonic Tomography

Ultrasonic sensor propagates acoustic waves within range of 20 kHz to over 1 GHz. Tomography system using ultrasonic sensor has major applications in medical field imaging (disease diagnosis, body parts visualization, fetal monitoring, blood flow measurements, therapy and surgery) and is slowly finding place in industrial flow measurement. In blood flow measurement, the sensor unit gathers flow

information by transmitting an ultrasonic signal into the blood and receiving the reflected Doppler-shifted ultrasonic signal (Christmann *et al*., 1990).

For flow concentration monitoring, ultrasonic imaging method uses two separate transducers, one for transmitting, and one for detecting the transmitted ultrasound on the contra lateral side of the object. As the suspended solids' concentration fluctuates, the ultrasonic beam is scattered and the received signal fluctuates in a random manner about a mean value (Mohd Hafiz, 2002).

In industrial applications, ultrasonic tomography is used to monitor and characterize the compositions of two liquid flows in a pipeline. An example of this is the water and oil mixture in oil palm industries. The advantages of applying ultrasonic tomography in industrial applications are non-invasive flow monitoring and imaging and non intrusive which can prevent the problem of pressure drop in the investigated vessel. The constraint in implementing ultrasonic sensors is usually the high cost of the sensors.

2.2.5 Nuclear Magnetic Resonance (NMR) Tomography

NMR or also known as Magnetic Resonance Imaging (MRI) is a radiology technique which applies the concepts of magnetism, radio waves and computer image processing to produce images of internal human body. Commonly used in medical field, MRI offers capabilities to look inside the human body, including the brain, heart, arteries and veins from any angle and direction without surgical invasion.

NMR is an emission technique based on stimulating radio frequency signals from nuclei by causing them to precess in phase in a strong magnetic field (Dyakowski, 1996). By inducing strong electromagnetic field, the protons of hydrogen atoms are aligned and exposed to beams of radio wave. This action spins the various protons of the body which eventually generates weak signals to be detected by the NMR scanner (Leong, 2004). Usually, signals are excited in a

selected two dimensional slice through the subject and positional information within this slice is obtained by superimposing a magnetic field gradient, so that nuclei in different regions precess at different frequencies (Dyakowski, 1996).

Compared to X-Ray, NMR offers higher contrast resolution, safer radiation and able to distinguish soft tissue in both normal and diseased states. However, NMR has the disadvantages of being very costly, large size and requires longer time to complete a scan. Industrial wise, NMR may be suitable for process monitoring which involves proton activities.

2.2.6 Optical Tomography

Optical Tomography has been developed by Abdul Rahim and Dugdale *et al.* based on an earlier investigation by Saeed *et al* in year 1988 (Rahim and Green, 1997). Optical tomography involves the use of non-invasive sensors to acquire vital information in order to produce two or three dimensional images of the dynamic internal characteristics of process systems (Ibrahim *et al.*, 1999). This technique is attractive since it is conceptually straightforward, relatively inexpensive and has a better dynamic response than other radiation based tomographic techniques.

Optical absorption happens when incident light energy passes through a medium and is converted to heat, causing attenuation. Different materials cause varying levels of attenuation and scattering and it is this fact that forms the basis of optical tomography (Ibrahim *et al.*, 2000). The working principle in Optical Tomography involves projecting a beam of light through a medium from one boundary point and detecting the level of light received at another boundary point. The analogue signals from each transducer vary in amplitude due to the received light being attenuated by particles passing through the beam (Rahim *et al.*, 1996).

Among the properties of interest for investigation in Optical Tomography are flow imaging concentration, flow velocity and mass flow rate. Optical Tomography can be applied in the pneumatic conveying in the industry of food processing, plastic

product manufacturing and solids waste treatment.

2.3 Overview of Embedded System

The rapid development in silicon die fabrication has led to the availability of high performance embedded systems which offer very competitive price, high speed operating frequency and low power consumption. Embedded systems are mostly suitable for applications that require low cost, small size and power saving features. A typical embedded system consists of the Central Processing Unit (CPU), supporting electronic circuits and firmware stored in its memory to instruct the CPU to carry out the intended tasks.

The operating frequency for a processor is usually measured in Million Instructions per Second or MIPS in short. Higher operating frequency allows processes to be performed faster but these devices usually cost higher. For two processors running at same frequency, they may not be executing instructions at the same rate. This is because some processors implement pipelining technology where a few instructions run in such a sequence that it seems that each instruction only requires one clock cycle. Without pipelining, the processor will execute each instruction in a few clock cycles before proceeding to the next instruction.

Usually, embedded systems are also equipped with many essential supporting peripherals such as Timers, Serial Communication (SCI), Serial Peripheral Interface (SPI), Event Managers, Analogue to Digital Converter (ADC), Digital to Analogue Converter (DAC) and Coder-Decoder (CODEC). The Serial Communication Interface (SCI) is the commonly used Universal Serial Asynchronous Receiver Transmitter (USART) for two wire communication. This protocol is very often used where low cost, robustness and simple communication is needed. It also gained popularity due to the widespread use of this protocol for devices such as mouse, simple data logger and modem on a standard personal computer. On the other hand, the Serial Peripheral Interface (SPI) finds place in communication between peripherals such as serial Electrical Erasable Programmable Read Only Memory

(EEPROM), serial ADC and communication between embedded systems. This serial communication protocol is often used to substitute the parallel bus architecture in order to reduce the number of data lines required at the price of lower data rate (Thiam *et al.*, 2004). This protocol is also called three wire serial interface and synchronous serial interface due to the fact that both devices that communicate share the same clock signal.

The ADC that comes together with an embedded controller simplifies hardware design and reduces the hardware cost compared to using dedicated external ADC. There are many available external ADC in the market and three important characteristics of ADC are conversion method, the number of bits representation and the rate of conversion. For the first characteristic, there are a few types of conversion methods, namely the Flash ADC, Successive Approximation Conversion (SAC) ADC, Tracking ADC, Single-slope ADC, Dual-slope ADC and Digital Ramp ADC (Floyd, 1997). The Flash converter converts analogue signal to its digital representation by using dedicated comparator for every bits it represent while the SAC converts analogue voltage by using a voltage comparison algorithm. Due to the extra comparators used in Flash converters, they are typically faster but cost more. The rate of conversion is also affected by the number of bits representation. Usually, 12 bits conversion is sufficient for most applications. For simple applications, 8 bits ADC can already do the job nicely. 16 bits ADC are usually used in applications where high accuracy is essential such as in music synthesizer and sound processing.

Memory plays an important role in embedded system applications. Usually, memory capacity is kept low in these systems in order to keep the cost low. Careful use of the memory is required so as not to waste the limited memory available. The memory comes in a few types such as EPROM, EEPROM, Flash memory and Static RAM. EPROM is programmable but in order to reprogram, the programmed device need to be exposed to the ultraviolet light for about fifteen minutes. The EEPROM eliminates this inconvenience by allowing programming using electrical signals. Flash memory offers higher speed in accessing the memory compared to EPROM and EEPROM but it comes with higher price. The Static RAM provides memory space for running programs to use in calculations since the memory access is fast.

Multiply and Accumulate or simply MAC is a relatively new technology in embedded systems. The use of this technology is driven by the need to perform multiplication and summation in many Digital Signal Processing algorithms. The MAC allows this process to be executed faster (typically 1 clock cycle) due to optimization being carried out at hardware level. Another hardware optimization that is available in some Digital Signal Processor is the Floating Point Unit (FPU). This unit is a dedicated processor responsible for floating point arithmetic processing. Storing floating point numbers in memory requires special format. In fixed point processors, software method is required to perform necessary conversions to store these floating point numbers. Using FPU, this method is not required.

2.3.1 Micro processors

Basically, micro processor is a digital integrated circuit that can be programmed with a series of instructions to perform specified functions on data (Floyd, 1997). It has three basic elements which are the Arithmetic Logic Unit (ALU), Register Unit and Control Unit. The ALU performs arithmetic operations such as additions and subtractions and logic operations such as NOT, AND, OR and exclusive-OR. The Register Unit stores temporary data during program execution while the Control Unit provides timing and control signals for performing programmed instructions.

Micro processors usually do not include memory device on chip. Memory devices such as Static Random Access Memory (SRAM), Electrical Erasable Programmable Read Only Memory (EEPROM) and Flash memory are usually wired externally to store data and programs. However, there are also some high end processors such as the processors for personal computers are equipped with memory blocks on chip. These memory blocks are usually small and are used as cache. Examples of micro processors are MC 6800 (Motorola Inc.), MC 68 000 (Motorola Inc.), PIC16F series (Microchip), PIC18F series (Microchip), 8051 (Intel Inc.) and 8x86 (Intel Inc.).

2.3.2 Micro controllers

Micro controllers are actually an embedded version of micro processors; where memory devices and other peripherals are embedded together in a single chip. This greatly simplifies the design and implementation of an embedded system since address decoding, memory interfaces and peripheral interfaces are all wired together in a single chip and ready to use. The overall cost is also reduced significantly since the cost to fabricate these devices during mass production at silicon level is lower than buying the peripherals chips separately and wiring them together. Besides that, the space for circuit implementation is also reduced since all peripheral chips are already in the micro controller.

However, the memory space provided in the micro controller is usually limitedly adequate for small programs. The operating frequency is commonly in the range of 4 MHz to 40 MHz, which is sufficient for general purpose control operations. Therefore, this type of embedded system is generally suitable for a single chip and small program operation such as in electronic toys, washing machine, microwave oven and digital radio. Examples of micro controllers are such as M68 HC11 (Motorola Inc), M68 HC12 (Motorola), AVR 8 Bit RISC (Atmel Inc.) and XA 16-bit series (Philips)

2.3.3 FPGA and ASIC

FPGA is an acronym for Field Programmable Gate Array while ASIC is an acronym for Application Specific Integrated Circuits. FPGA consists of a matrix of reprogrammable logic OR and logic AND gate arrays which defines the functionality of the FPGA. Widely used computer architectures have a Central Processing Unit (CPU) that operates sequentially on data stored in memory based on the program loaded. The instructions in a program are executed one by one sequentially by the CPU at each clock cycle. This is disadvantageous for algorithms which can be executed in parallel. In such case, FPGA can be implemented in order to perform parallel data processing where more than one tasks can be carried out concurrently. This is possible since FPGA is really a ‘wired hardware’. FPGA is usually used for

Application Specific Integrated Circuit or ASIC in short, for prototyping due to FPGA reprogrammable feature and low cost.

On the other hand, ASIC is a technology where custom designed circuits are fabricated at chip or die level. As a result, this chip is usually not reprogrammable. Usually, this technology is chosen in cases where the chip will be mass produced and there will be no change or further upgrade to its firmware.

2.3.4 Digital Signal Processors (DSP)

DSP is a hardware technology mainly implemented to optimize the processing of algorithms used in digital signal processing. Signals such as voice and music were impossible to be processed digitally in real time without the use of DSP. In other applications, image processing such as used in digital camera and camcorder also utilizes DSP. Typically, DSP is used in combination with micro processors or micro controllers where DSP handles the signal processing and behaves like a co-processor.

DSP can be categorized in two main groups which are fixed point processor and floating point processor. A fixed point DSP can only store and process integer values without special software while floating point DSP can store and process decimal values with its additional supporting hardware. Floating point operation is still possible for fixed point processor in the condition that the firmware converts between the floating point value and the value stored in memory using a conversion algorithm. This takes up a lot of processing time from the processor. In floating point type DSP, there is a Floating Point Unit (FPU) to handle this conversion at hardware level. The processor can then concentrate its processing on the main tasks. Because of this extra hardware, the cost for floating point DSP is relatively higher than the fixed point DSP.

In comparison to other types of microprocessors, DSP processors often have an advantage in terms of speed, cost, and energy efficiency. In comparison with

conventional analogue processing devices, DSP offers easily configured systems and the digital components are widely available at relatively lower cost than high quality analogue devices. Another advantage of using DSP is that it is programmable so that any changes can be made on the fly. In comparison, analogue parts are usually bulky and more expensive than their digital counterparts.

These attractive features motivate DSP implementation in tomography system where programmable, high speed capability to support intensive computation and low cost processor is required. DSP has gained its place in medical field tomography applications where previous work by Yuan and Yu (2003) implemented DSP as the main processing unit in EIT system for real-time image monitoring in medical application.

2.4 Mass Flow Rate Measurement

The vast majority of flow meters provide a measure of volumetric flow. For fuel flows, it would be preferable to measure mass flow as this is generally proportional to energy flow. Very few flow meter types provide such a reading while the most common ones are thermal mass, angular momentum transfer, and coriolis meters (Guilbert *et al.*, 1996)..

Mass flow rate measurement can be divided in two main groups which are the true mass flow rate and the inferential mass flow rate. A true mass flow meter has a sensing element that reacts to the mass flow of solids through the instrument. On the other hand, an inferential mass flow meter determines both the instantaneous solids concentration and velocity within the sensing element and multiplies these two values to give the mass flow rate (Beck, 1987).

At present, there are mainly two methods used for measuring solid mass flow rate in a conveyor. The first method is the calibration based method, whereby a measuring instrument is calibrated on a flow rig using the actual solid material. This calibration process often involve plotting a graph of low, medium to high flow rates

against the volume concentration (Chan, 2002). The volume concentration can then be used to determine the solids mass flow rate.

The second method of measuring solids mass flow rate is by using algorithms, whereby calibration constants are not required. Xie *et al.* (1989) presented a method to measure the solid mass flow rate by measuring two parameters, namely the solids velocity and the volumetric concentration as in Equation 2.1

$$W_s = A_0 \rho_s \beta U_s \qquad ...2.1$$

Where,

W_s = Solids Mass Flow rate (kg s^{-1})

A_0 = Cross sectional area of the pipeline (m^2)

ρ_s = Mass density of the conveyed solids (kg m^{-3})

β = Normalized volumetric concentration of the solids ($0 \leq \beta \leq 1$, dimensionless)

U_s = Solids velocity (m s^{-1})

Pang (2004a) in his research combined A_0 and β as an average concentration area, using an area sensitivity map per pixel. This resulted in a mass flow rate per pixel, namely the mass flow rate profile. The mass flow rate profile is averaged to obtain average mass flow rate.

Calibration based mass flow rate measurement requires less processing and therefore can achieve higher throughput. The disadvantage is that it requires calibration work which is time consuming whenever there is change of the solids flow material or the flow rig system. On the contrary, algorithm based mass flow measurement requires high performance processing capabilities to support the large amount of calculations involved in the algorithms. The advantage is that it does not require any calibration work prior to measurement Pang (2004a).

2.4.1 Concentration Profile Measurement

One of the main aims of Process Tomography is to view the flow concentration of the flowing material in a pipeline conveyor. The cross sectional image is obtained by using image reconstruction algorithms. There are many algorithms to reconstruct the cross sectional image, but the most basic and simplest form is the Linear Back Projection reconstruction algorithm. This algorithm back projects the sensor readings in order to locate the flow location and therefore a cross sectional image. Modifications made to this algorithm by Ibrahim *et al.* (1999) resulted in Hybrid Reconstruction algorithm which filtered out smearing effect and ambiguous images.

There are also other image reconstruction techniques which aim to solve many other problems related to performance and accuracy of the reconstructed image compared to the actual model. For instance, Iterative Image Reconstruction algorithm aims to provide more accurate image by reducing the image error at each iteration. On the other hand, Fourier Slice Theorem reconstruction algorithm enables high accuracy image to be reconstructed but requires a large number of projections at different angles (Kak and Slaney, 1999).

2.4.2 Velocity Profile Measurement

In tomography applications, flow velocity is an important flow characteristic for analysis. Excessively high particle velocity will cause high energy consumption, severe pipeline wear and particle degradation (Yan, 1996).

There are mainly two groups of methods to obtain flow velocity which are the Doppler methods and tracer methods. The Doppler methods applies Doppler shift principle using electromagnetic radiation source such as laser, microwave or ultrasonic as energy source (Thorn *et al.*, 1982). The laser method is expensive and fragile for routine velocity measurements in pneumatic conveyors while ultrasonic method has difficulty in obtaining efficient energy coupling into the gas flow from

the ultrasonic transmitter and also susceptible to errors due to changes of temperature of air conveying the solids (Thorn *et al.*, 1982).

There are two variations of tracer methods. The first is by injecting recognizable markers (usually radioactive) into the flowing solids stream and their movement progress timed between two points. Velocity can be calculated from the distance of the two points and the travelling time of the marker. The disadvantage of injecting radioactive markers is contamination and the contaminated bulk material is wasted. The second method is by using natural disturbances of the flow to generate 'flow noise' for cross correlation (Thorn *et al.*, 1982). Cross correlation technique offers a cheap, robust and safe method to measure flow velocity.

An assumption for using cross correlation technique for flow velocity measurement is that the arrangement of particles flowing through the downstream sensor layer is a time-delayed replica of the upstream sensor layer (Ibrahim and Green, 2002a). Although the arrangement will change slightly in practice, there is still a recognizable part of the pattern conveyed between the sensors.

Foster *et al.* (1990) investigated time domain cross correlation with ultrasonic sensors and came up with conclusions that there were a few factors that affected the measurement precision. They are the beam width, data acquisition window length and range of velocity gradient to be captured.

2.4.2.1 Cross Correlation Technique

The principal of cross correlation flow measurement uses the random fluctuations in the gas or solid distribution within the fluid to sense flow (Byrne *et al.*, 1990). Mathematically, cross correlation is used to find out the similarity between two signals and is shown as in Equation 2.2.

$$R_{xy}(\tau) = \frac{1}{T}\int_0^T x(t)\, y(t-\tau)\, dt \qquad \ldots 2.2$$

Where,

$R_{xy}(\tau)$ = Value of cross correlation function when the upstream signal is delayed by a time of τ

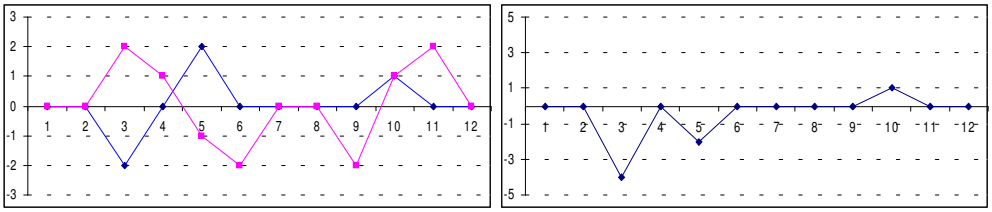
x(t) = Downstream signal

y(t) = Upstream signal

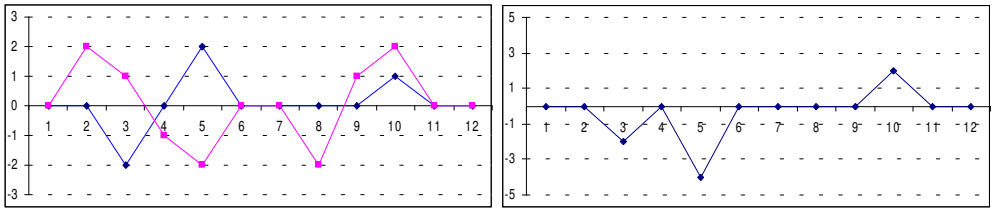
Usually, this technique carried out by shifting one signal towards the other and then multiply-accumulate the two signals. Results from the multiplications are interpreted by considering the peak value, where the number of shifts that causes the peak value is the number of sample delay between the two signals (Ibrahim and Green, 2002a). This concept can be applied in velocity measurement in tomography system where the delay is the time for the object to travel between two sensors separated by a distance.

By performing manual graphical analysis (illustrated in Figure 2.2), it can be seen that cross correlation in time domain actually involves point to point multiplication of the two signals, where one of them (x_1) is fixed at one location while the other signal (x_2) shifts with time. The multiplication result is then summed up as cross correlation coefficients (r_{12}). For example, consider a signal being cross correlated to an identical but time delayed signal. The sum of point to point multiplication between the two signals will be the highest when the shifted signal is being shifted back to the original location (refer Figure 2.2, Lag = 6). Therefore, the number of time shifts needed to obtain the peak coefficient is actually the time lag between the two signals.

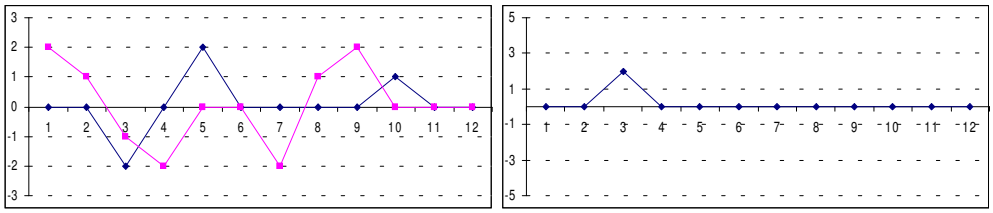
Lag = 0 x_1 = blue, x_2 = magenta $r_{12}(0) = \sum(x_1 * x_2 (n\text{-}Lag)) = -5$



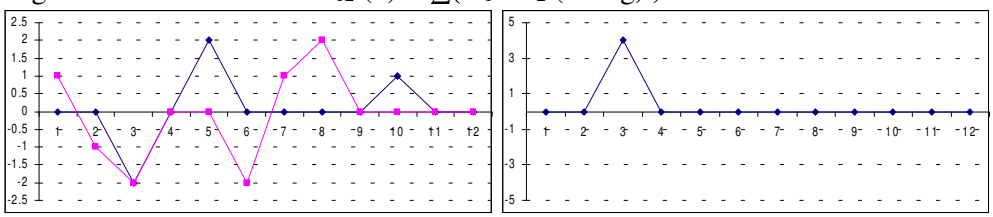
Lag = 1 $r_{12}(1) = \sum(x_1 * x_2 (n\text{-}Lag)) = -4$



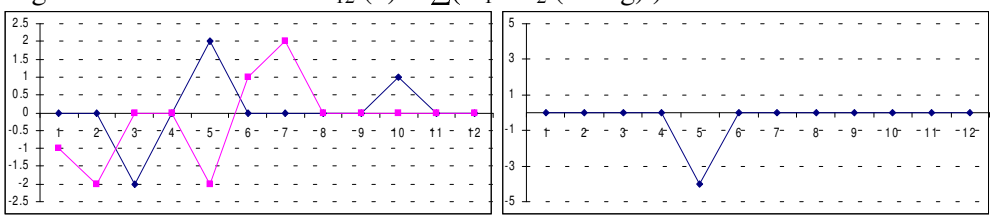
Lag = 2 $r_{12}(2) = \sum(x_1 * x_2 (n\text{-}Lag)) = 2$



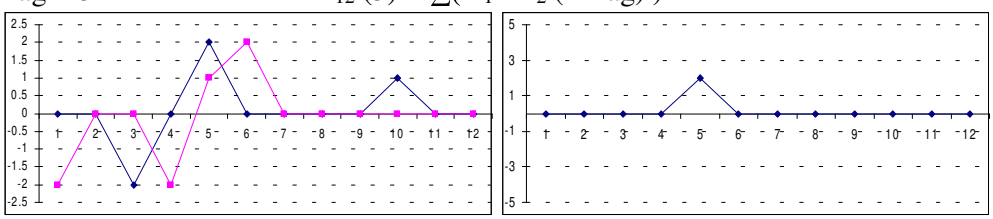
Lag = 3 $r_{12}(3) = \sum(x_1 * x_2 (n\text{-}Lag)) = 4$



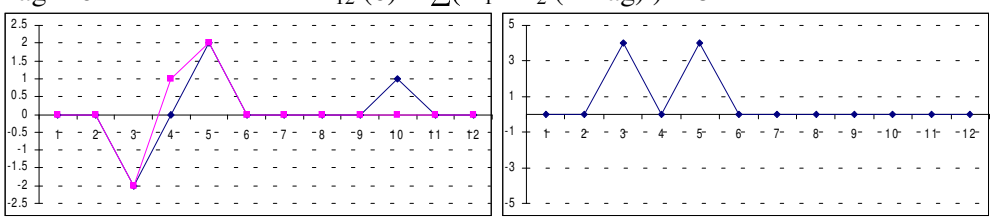
Lag = 4 $r_{12}(4) = \sum(x_1 * x_2 (n\text{-}Lag)) = -4$



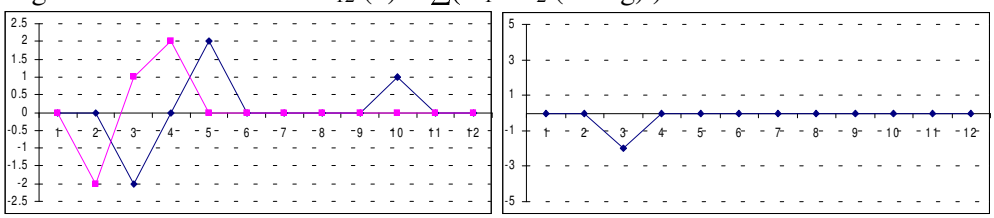
Lag = 5 $r_{12}(5) = \sum(x_1 * x_2 (n\text{-}Lag)) = 2$



Lag = 6 $r_{12}(6) = \sum(x_1 * x_2 (n\text{-}Lag)) = 8$



Lag = 7 $r_{12}(7) = \sum(x_1 * x_2 (n\text{-}Lag)) = -2$



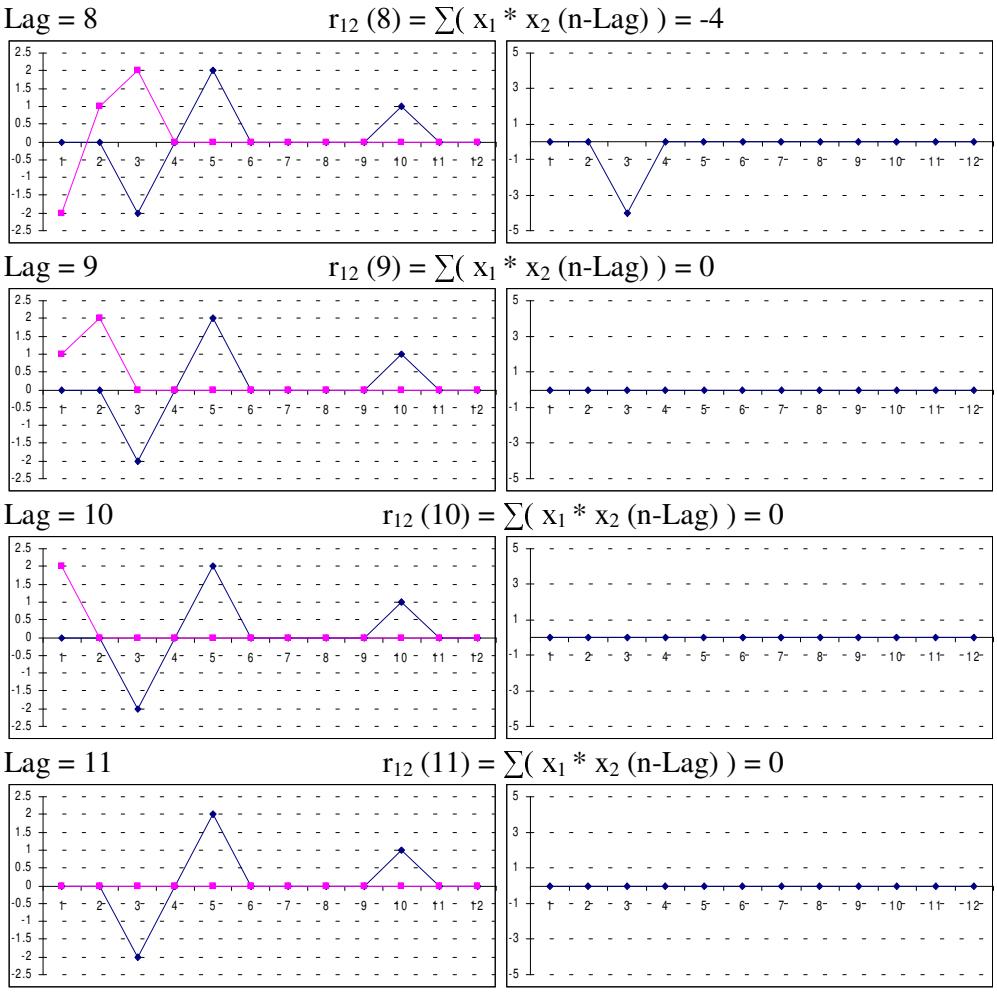


Figure 2.2 : Graphical illustration of cross correlation function

Based on assumption that the signal going through the downstream sensors are time delayed replicas of the signal going through the upstream sensor (Ibrahim and Green, 2002a), high similarity between the two signals can be determined by using cross correlation function. The result from the cross correlation function is the time delay between the first and the second signal. In the case of process tomography system, this time delay translates to the period of an object to fall from the upstream sensor to the downstream sensor. Since the distance between the upstream sensor and downstream sensor is known, the average velocity of the object passing through can be calculated by Equation 2.3 (Thorn *et al.*, 1982).

$$v = \frac{L}{t_d} \qquad \ldots 2.3$$

Where,

v = Object velocity

L = Distance between upstream and downstream sensor

t_d = Time delay from cross correlation function

It should be mentioned that t_d is the time delay and cross correlation only results in the number of time delays (lags) which is dimensionless. The sampling time or the period between each sample in the signal series should be multiplied to the number of lags to obtain t_d. This is shown in Equation 2.4.

$$t_d = t_s L \qquad ...2.4$$

Where,

t_d = Time lag between upstream and downstream sensor

t_s = Sampling time

L = Number of lags / shifts for maximum cross correlation value

The cross correlation function was carried out after image reconstruction, where each upstream pixel was correlated to its respective downstream pixel to determine the time delay. This method was termed Pixel to Pixel Correlation. Similar process when carried out to signals from upstream sensor values and downstream sensor values was termed as Sensor to Sensor Correlation (Ibrahim and Green, 2002a).

2.4.2.2 Pixel to Pixel (PTP) Correlation

PTP Correlation method requires that image reconstruction algorithm to be performed first on the sensor data acquired. At least a hundred frames of data (each set containing a single tomogram) from upstream and downstream sensor layer need to be obtained for cross correlation in order to have reliable correlation (Mohd Hezri, 2002). Then, each upstream pixel is correlated with its respective downstream pixel to obtain the lags. Next, the lags are converted to time by multiplying them with the sampling time (data acquisition period of each frame). Thus, the time delay now

contains the period that the object takes to fall from upstream to downstream sensor. Finally, the distance between upstream and downstream sensor is divided by the time delay at each pixel to obtain the velocity profile as in Equation 2.2 shown previously. The challenge of implementing this technique is the large number of cross correlation to be performed, which is N x N times for N x N orthogonal sensor pairs.

2.4.2.3 Sensor to Sensor (STS) Correlation

STS Correlation method uses upstream sensor data to correlate with downstream sensor data instead of using pixel values (Ibrahim and Green, 2002a). The advantage of this method is that the number of cross correlation required is only 2 x N times for N x N orthogonal sensor pairs. This number is N/2 times less than that required by Pixel to Pixel Correlation. The disadvantage of this method is it only obtains a single periphery velocity per sensor pair.

In this research, studies were carried out to manipulate this technique in order to obtain the velocity profile.

2.5 Summary

This chapter reviewed the typical tomography systems in terms of working principle, advantages, disadvantages and applications of these systems. This was followed by an overview of embedded system with focus on the types of embedded systems and the supporting hardware peripherals typically available on embedded systems. Then, review of solids mass flow rate measurement techniques which consists of the concentration profile measurement, velocity profile measurement and cross correlation technique were presented.

CHAPTER 3

HARDWARE SYSTEM DESCRIPTION

3.1 Introduction

Typical hardware setting of an optical tomography system consists of an optical sensor array, a data acquisition unit, a processing unit and a result display unit. The sensor array (photo diode or photo transistor) converts the parameter of interest which is the light intensity emitted from an array of light (usually LED), into small electrical signals. Experimentally, it has been shown that the relationship between the number of solids passing through a beam and the corresponding sensor output voltage is linear (Ibrahim *et al*., 1999). These signals will be digitized by a data acquisition system and logged for processing. Following that, a data processing unit will be used to analyse the data in order to produce results that describe the flow behaviour including flow concentration, flow velocity and mass flow rate. Finally, the results will be displayed on a monitor and a LCD panel.

In this research, the hardware development was divided into three main modules, as follows:

i. Signal Conditioning Unit
   - Analogue Signal Conditioning Circuit
   - Digital Signal Conditioning Circuit

ii. Main Controller Unit
   - Embedded system controller

iii. User Interface Unit
   - Embedded system controller

- 2 line, 16 character LCD
- 4 x 4 matrix keypad

The topology is illustrated in the following figure:

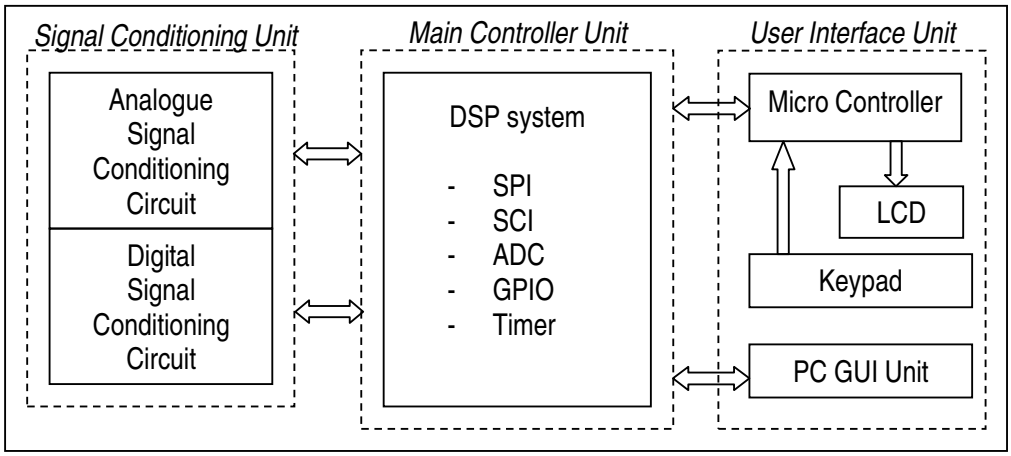


Figure 3.1 : Block diagram of three main hardware modules

The first module which is the Signal Conditioning Unit is responsible for the conditioning of incoming signals from the optical sensors. For the Analogue Signal Conditioning, the incoming signals are from 64 optical sensors of the orthogonal layers. These signals are multiplexed into 16 ADC channels on the DSP system. On the other hand, 92 single bit binary streams from the Digital Signal Conditioning are sent to the embedded system using Serial Peripheral Interface (SPI). There are also signals from the DSP system to control the sampling process of these two modules.

The Main Controller Unit as its name implies, acts as a main controller in processing the measurement data and producing required results. It is mainly used to perform thousands of calculations required by image reconstruction algorithm, velocity profile construction and mass flow rate measurement. It also coordinates the communication between Signal Conditioning Unit and User Interface Unit.

3.2 Overview of Sensors Configuration

In this research, a fabricated optical sensor unit with sensors arranged in parallel beam projection has been implemented (Pang, 2004e). There are two layers of sensors being used, namely Upstream Layer and Downstream Layer to support the analysis of algorithm-based velocity profile measurement. At each layer, there are four arrays of sensors arranged in parallel beam projections. Two arrays of sensors, namely Orthogonal Sensors are arranged in such a way that their projected light beams are orthogonal (90º) to each other while another 2 arrays of sensors namely, Rectilinear Sensors are arranged so that their projected light beams are 45º to the orthogonal projections. This configuration is illustrated in the following figure.

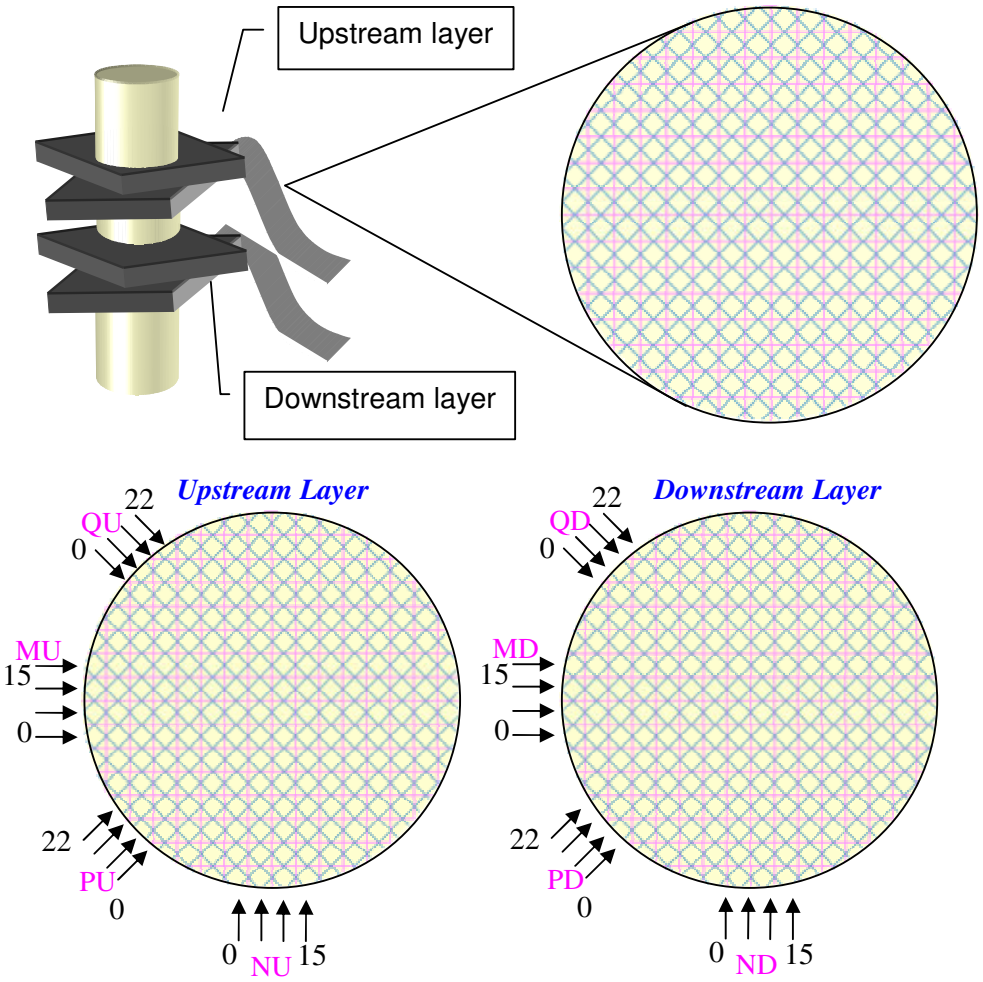


Figure 3.2 : Sensor Unit configuration

There are 16 sensors at each orthogonal projection (MU_0–MU_{15}, NU_0–NU_{15}, MD_0–MD_{15}, ND_0–ND_{15}) where altogether is a total of 64 sensors. The suffix U and D appended were to represent the Upstream and Downstream layer. Another four arrays of 23 sensors (PU_0–PU_{22}, QU_0–QU_{22}, PD_0–PD_{22}, QD_0–QD_{22}) are arranged to provide rectilinear projections. This adds up 92 sensors for the rectilinear projections for Upstream and Downstream layers. In total, there are 156 sensors readings to be acquired at every single scan.

3.3 Signal Conditioning Unit

The Sensor Unit already includes LED driver circuits and amplifier circuits to convert the received light to electrical signal. Each sensor from the Sensor Unit outputs a current which is inverse proportional to the light intensity it measures. In other words, higher current is output when there is little light (object presence blocks the light beam) and lower current is output when there is more light (no object blocking the light beam). This way, software overhead is reduced during image reconstruction as the presence of object is proportional with the voltage measured. In work by Chan (2002), optical attenuation model was used, where object presence resulted in voltage loss. The processing software then subtracted the loss from the expected full voltage to obtain the correct readings. This step was not required in this research.

All the signal amplifiers on the Sensor Unit are identical. Only resistors are required to be added to the outputs of the Sensor Unit to have a voltage drop for measurement. The circuit for a single sensor is shown in the following Figure 3.3.

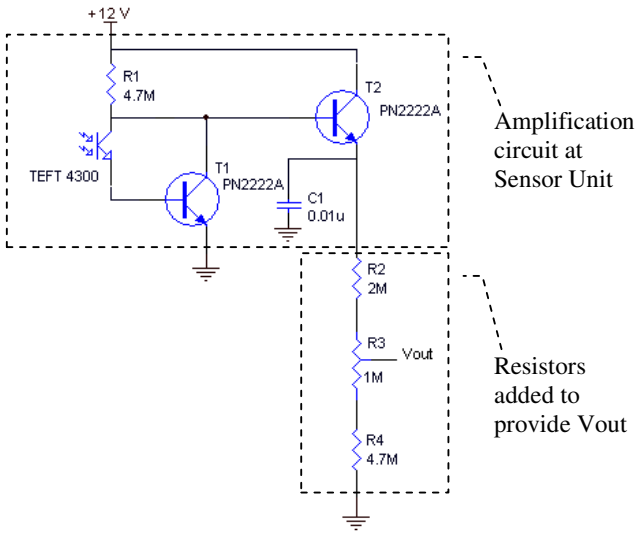


Figure 3.3 : Amplification circuit for each sensor

In designing the signal conditioning circuits, there are two considerations that need to be observed. The first consideration is the data acquisition rate. In total, there are 156 sensors readings to be acquired at every single scan. This obviously causes the measurement to take very long time, especially using a single digitizer with low conversion rate. Using a low conversion rate digitizer, an array of sample and hold circuit is required to acquire and hold the readings of a single scan (Pang, 2004a). This is because the conversions occur sequentially and by the time the digitizer converts the next sensor, the current signal is not from the same time as captured by the previous sensors. This may result in tilted or distorted image. Since there are 156 sensors on the sensor unit, using sample and hold circuits for each sensor will also take up a lot of board space.

Another approach is to speed up measurement time by using dedicated low conversion rate digitizer per sensor or per group of sensors. The digitizers can be triggered to convert many signals simultaneously so that the need of sample and hold circuits can be eliminated. However, this approach is expensive and still occupies a lot of board space.

As a solution, a high conversion rate ADC was implemented in this research to overcome the said issues. The use of a single high conversion rate ADC required

analogue multiplexers to multiplex respective signals for conversion. In this research, the ADC being used was built-in inside the DSP chip. Internally, it was multiplexed to receive analogue signals from 16 external channels. Since there are more than 16 signals for conversion, additional analogue multiplexers still need to be designed.

As for the second consideration, a faster method compared to ADC for sampling the sensors on rectilinear projections is desired. This is because these sensors only provide masking of ambiguous image in Hybrid Reconstruction algorithm. Previously, these signals were converted using ADC and compared to a threshold value set in software (Pang, 2004a). If the measured signal value was above the threshold value, a TRUE value was obtained and vice versa. Then, the signal value was discarded. Therefore, a single bit binary representation is sufficient and this can be implemented using hardware voltage comparator which can perform this same task at a higher rate. This approach reduces the number of signals for ADC conversion from 156 signals to only 64 signals.

3.3.1 Analogue Signal Conditioning Circuit

Basically, the Sensor Unit has provided the amplifications to convert the light intensity to current. Resistors were added to the output of the amplifier circuits to generate voltage drop for measurement. The fabricated circuits on the sensor unit were based on optical path width model (Pang, 2004a) where the current outputs correlate linearly with the width of the object that obstructs the light beam.

As mentioned previously, there are 16 sensors per projection and there are two orthogonal projections per layer. Therefore, there are 32 signals for Upstream Layer and 32 signals for Downstream Layer, totalling up to 64 signals for measurement. In this research, a built in ADC in a DSP of the Main Controller Unit was utilized to perform the signal acquisitions. The specifications of the ADC are as stated in the following table:

Table 3.1: ADC specifications

Parameter	Value
Number of bits	12 bit
Analogue Input	0 – 3 volts
Conversion time	80 ns / 12.5 MSPS
Channels	16
Total Result Register	16
Auto sequencer Support	Yes

The ADC is multiplexed into two groups, namely Group A and Group B. Each group consists of eight channels, thus expanding the total channel inputs of the ADC to be 16 channels. Due to the fact that there are 64 signals to convert, the signals need to be multiplexed. Four dual 4 to 1 analogue multiplexers (4052) were used per layer to multiplex these signals. Using this configuration, all 64 channels measurements can be accomplished in four acquisitions. The dual 4 to 1 analogue multiplexer was also chosen instead of the 8 to 1 analogue multiplexer since there were 16 ADC channels that were available. The data acquisition process can also be controlled easily since only two input bits (Sel MSB and Sel LSB) need to be controlled by using GPIO port. The overall configuration is shown in the following figure.

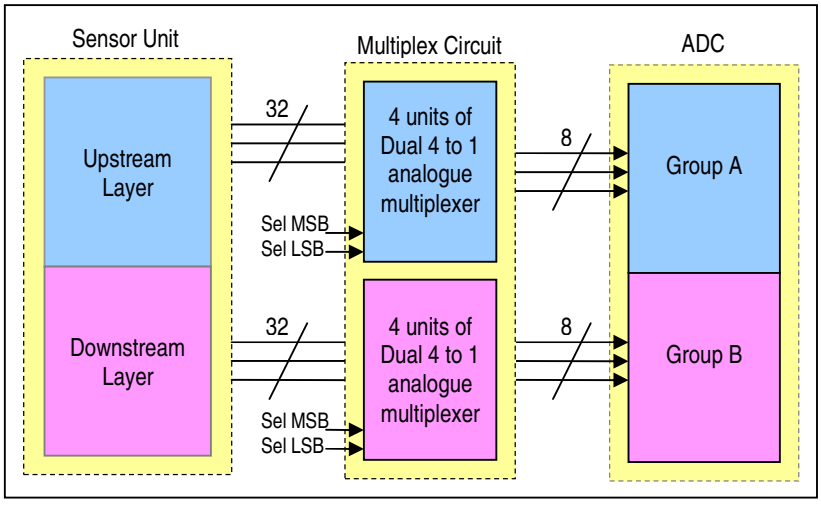


Figure 3.4 : Interface between analogue signal conditioning and ADC

3.3.2 Digital Signal Conditioning Circuit

As discussed in Section 3.3, improvement on the sampling method of the sensors for rectilinear projections was implemented using hardware voltage comparator. This fulfils the requirement of the image reconstruction algorithm where only a single bit representation is required (Either TRUE or FALSE). Each voltage comparator is implemented using TLC 2354 referenced to a voltage divider using variable resistor. These variable resistors were calibrated to a threshold value of about 25% of the steady state sensor output voltage (4V). This made sure that the representation was valid and not caused by false trigger such as noise and spikes. The following Equation 3.1 represents the value of the binary bit for different sensor voltage levels.

$$Vout = \begin{cases} LOW & ,Vin < 1\,V \\ HIGH & ,Vin \geq 1\,V \end{cases} \qquad \ldots 3.1$$

The circuit for each voltage comparator is shown in the following figure.

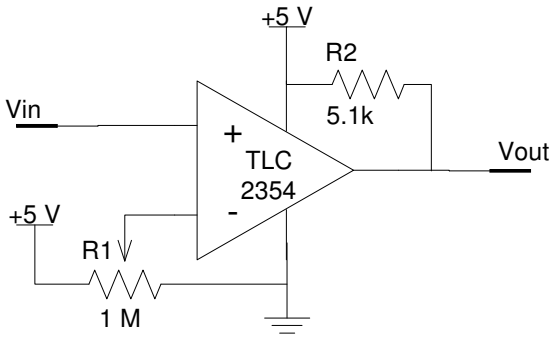


Figure 3.5 : Voltage comparator circuit

After the voltage comparisons were carried out, the binary bits need to be sent to the embedded system. This was conveniently achieved by converting them to serial bit stream and then reading the data using SPI. Each sensor output from the rectilinear projections was fed to a voltage comparator input (Vin). The resulting outputs (Vout) from the comparators were then connected to the data inputs of

parallel to serial converter chips (74HC165) arranged in cascade. The embedded system only needed to load the parallel data bits by sending a pulse to the Parallel Load pin and generate clock signal to shift in the serial data. From the embedded system side of view, the Digital Signal Conditioning Circuit is merely a slave device with serial interface. This approach is less troublesome compared with using bus topology where buffers and more digital input and output lines are needed (Thiam *et al.,* 2004). Besides that, the chip is able to support high speed data transfer. The overall topology of the digital signal conditioning circuit is shown in the following figure.

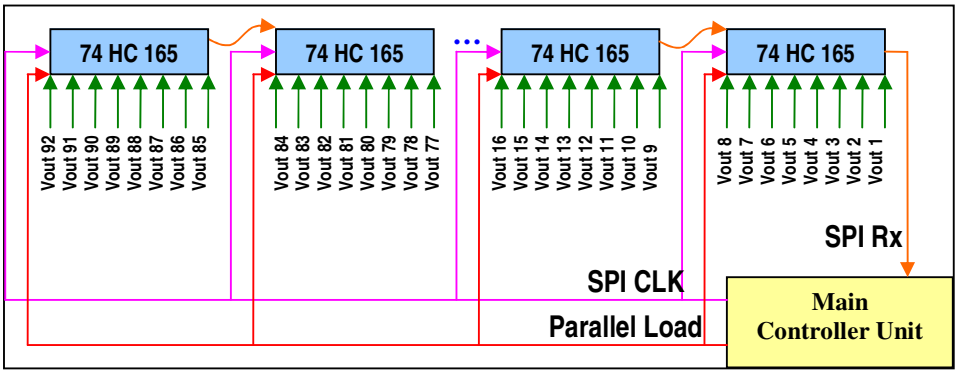


Figure 3.6 : Digital Signal Conditioning topology

3.4 Main Controller Unit

The main controller unit is the brain of the whole system. The processing involved includes data acquisition, image reconstruction from measured sensors signals, flow analysis in terms of flow velocity and mass flow rate measurement and communication between modules. All these processes require intensive calculations and thus, the performance of this controller is very essential in affecting the overall system performance. Pang (2004b) developed an optical tomography that could perform image reconstruction in real time by using data distribution system where 3 sets of high performance PC (2.4 GHz, 512 MB RAM) were used. This research attempted to achieve comparable result using only single embedded system as its main controller unit.

Among the criteria that need to be fulfilled by the embedded system are high execution speed, low cost, support for standard communication protocols and memory expansion capability (Pang, 2004). Micro processor based systems can offer high flexibility but requires more complicated design and other peripheral interfaces to support required functionalities. On the other hand, micro controller based systems offers many built in and ready to use standard peripherals but are usually not very flexible. Finally, a Digital Signal Processor based system provides relatively greater performance but lacks in terms of interfacing to the outside world.

A relatively new technology was introduced to the market lately and was known as the Hybrid Digital Signal Processor. The hybrid term was actually a combination of the micro controller and Digital Signal Processor technologies. As such, this architecture is actually a solution that benefits the best of both worlds. As this research requires both the peripherals offered by micro controller and high speed processing offered by Digital Signal Processor, the Hybrid Digital Signal Processor makes a very suitable choice as the platform for implementation in this research.

Table 3.2 : Comparison of available processor

Family	**Core Clock Speed (MHz)**	**Remarks**
TMS320F28xx	150	Hybrid DSP, built in peripherals : UART (2), SPI, CAN 2.0, ADC (16.7 Msps), 128k Flash memory, GPIO, Dual 16x16 bit MAC
TMS320C55xx	300	Dual MAC, DMA, GPIO , Host Processor Interface (HPI), UART, ADC, I^2C,USB Full Speed (12 Mbps)
TMS320C64xx	600	Host Processor Interface (HPI), UART, GPIO (16 bits), 20 bit video port

A comparison was carried out on some of the Digital Signal Processors offered by Texas Instruments Inc. (Texas Instruments, 2002i) and the results are shown in Table 3.2. It can be concluded that the most suitable processor for this research is the one from the TMS320F28xx family. A device from this family, TMS320 F2812 fulfils almost all of the requirements needed. The unavailability of

the Floating Point Unit is considered as a trade off for its low cost and can be justified by many of the supporting peripherals that come together in the chip.

3.4.1 Digital Signal Processor Architecture

The chosen DSP (TMS320 F2812) as the main controller in the embedded system is a well equipped architecture with many essential peripherals built in and ready to be used. The following figure shows the block diagram of the overall architecture in the F2812 processor (Texas Instruments, 2002j).

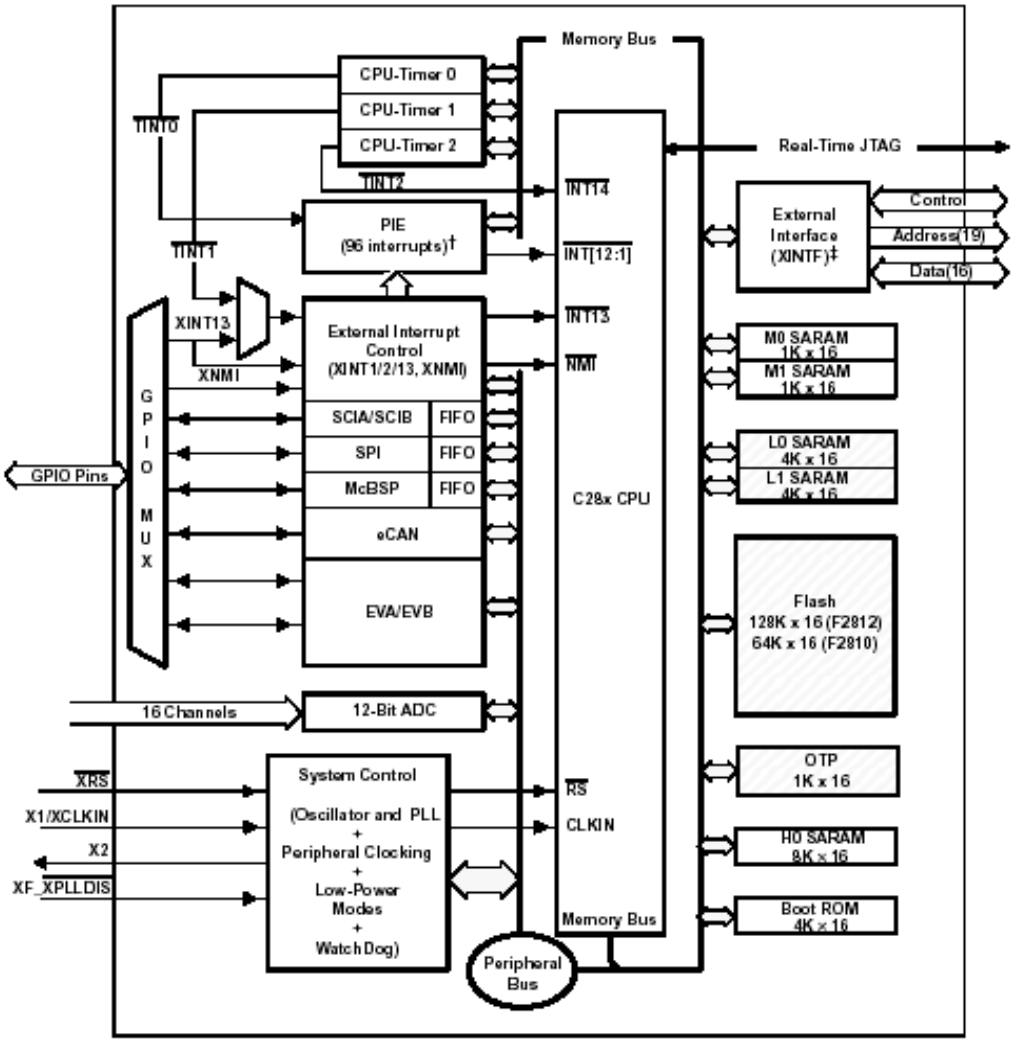


Figure 3.7 : TMS320 F2812 hardware architecture

3.5 User IO Unit

As mentioned previously, this unit allowed user to view Mass Flow Rate result and change certain settings of the Main Controller Unit. This unit consists of a microcontroller, LCD module and a 4x4 matrix keypad. By implementing this module, the control of LCD and keypad processing became transparent to the Main Controller Unit. Hence, the Main Controller Unit was able to communicate with this module using only the serial communication interface.

3.5.1 The Microcontroller Architecture

The User IO Unit was developed using M68HC11 microcontroller from Motorola Inc. This microcontroller which is commonly known as HC11 contains many built in peripherals such as SCI, SPI, RAM, EEPROM, ADC, GPIO and Timing modules. It is typically driven by an 8 MHz crystal oscillator, which provides the main clock of 2 MHz (the main clock is divided by four). For LCD display and keypad handling purpose, this speed is already sufficient. The HC11 architecture is shown in Figure 3.8.

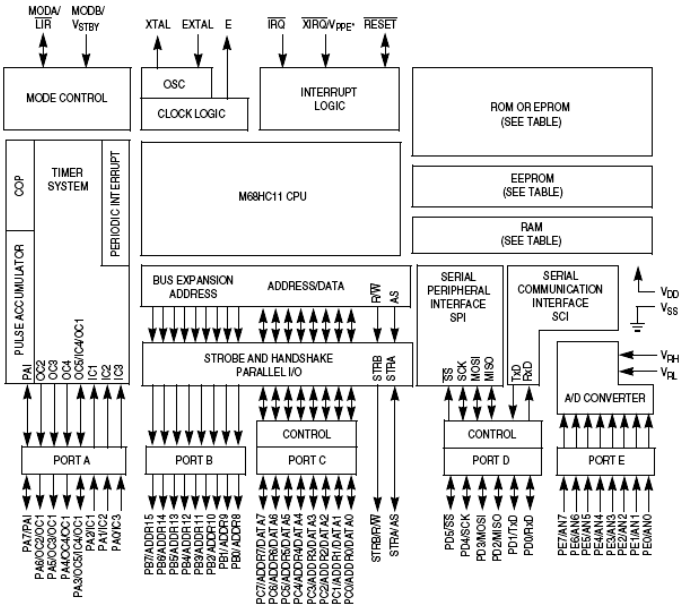


Figure 3.8 : HC11 hardware architecture.

Shown in Figure 3.9 is the circuit diagram for the User IO Unit. Basically, the circuit is a minimum system that uses its GPIO pins to drive the necessary signals to the LCD module and decode the keypad. The LCD driving algorithm and keypad decoding algorithm will be explained in Chapter 4.

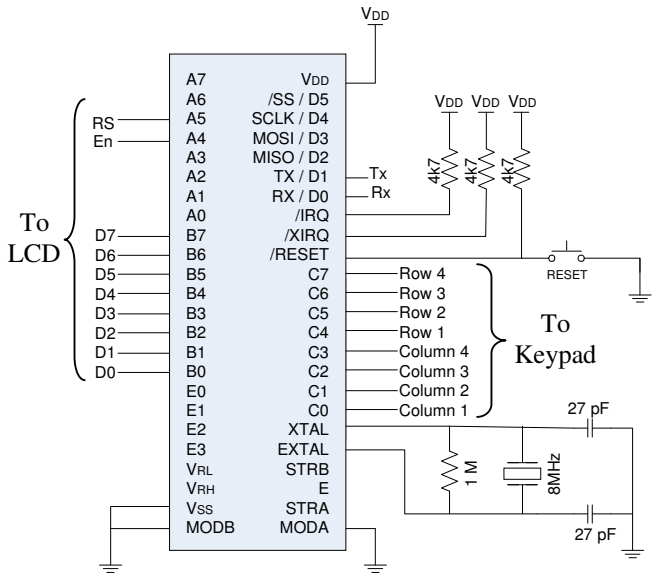


Figure 3.9 : User IO Unit circuit diagram

3.5.2 Liquid Crystal Display (LCD)

The Liquid Crystal Display is a standard 2 lines, 16 characters alphanumeric LCD module. The alphanumeric display is actually driven by a LCD driver built-in in the module. The LCD driver contains a table of standard alphanumeric characters stored in its volatile memory. Commands and data are sent to this driver by driving the necessary pins to certain patterns. The following figure shows the connections of the LCD module to the embedded system.

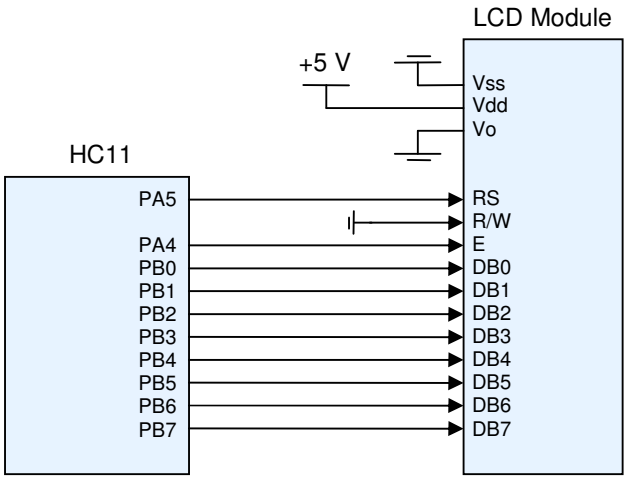


Figure 3.10 : Interfacing LCD module with HC11

The functions of the pins are shown in the following table.

Table 3.3 : LCD module pins functions

Pin	Function
Vss	Power ground
Vdd	Power +5 V
Vo	Power supply for LC drive (optional)
RS	Register Select - choose Instruction or Data
R/W	Read / Write Instruction
E	Enable Module
DB0 - DB7	Data bus (8 bits)

In order to display data to the LCD module, standard protocol and commands set were used. These commands are usually standard among most LCD modules. Basically, the microcontroller needs to handle the control signals according to certain timing requirements (RS, 2001) and outputs the command or data to the data bus accordingly. An example of the signals timings is shown in Figure 3.11.

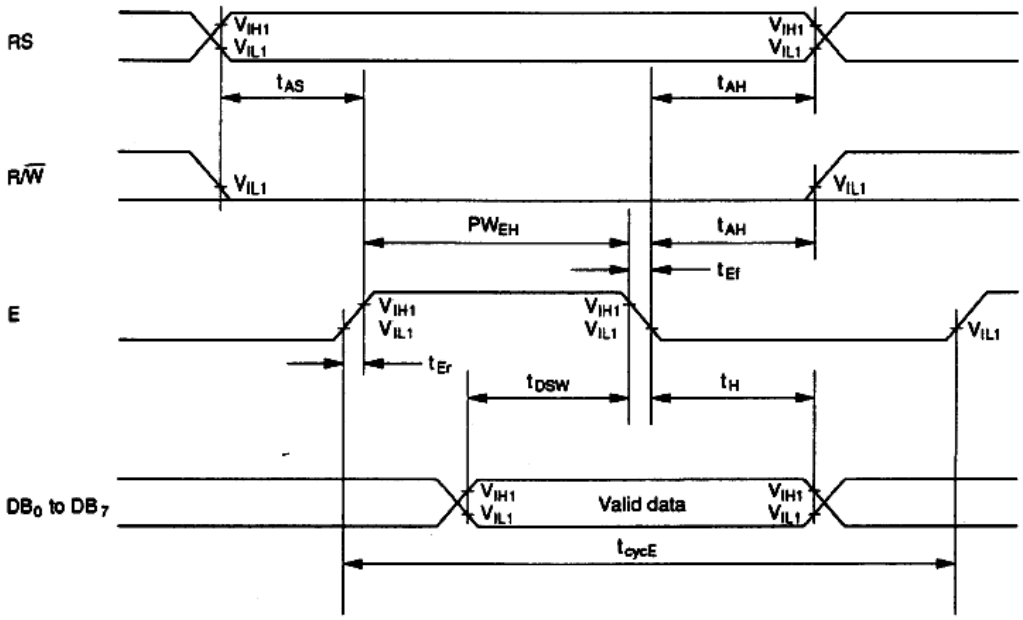


Figure 3.11 : LCD write mode signal timing

3.5.3 Keypad

The main usage of the keypad is as the input interface for the user when the PC GUI unit is not being used. Settings such as solid density and status of PC GUI unit (online or offline) are inputs that can be entered through this interface. The keypad used in this research is a standard 4x4 matrix keypad. It contains numbers from zero to nine and characters from A to D (Shown in Figure 3.12). This interface acts as an input to user to change settings if required. The microcontroller is responsible to decode the key pressed by the user and send it to the Main Controller Unit.

This 4x4 matrix keypad can be decoded easily using software technique. Basically, the keypad is divided to 4 rows and 4 columns. Each column is pulled high with a pull up resistor. At every scan, the first row is set to logic LOW while the other rows are set to logic HIGH. Then, all columns logic states are scanned. If a key is pressed, it will cause that particular column to be at logic LOW. This process is repeated with the other rows set to logic LOW until a key press is detected. Then, the value of key pressed is sent to the LCD module and also to the Main

Controller Unit. The Main Controller Unit will receive the key pressed and update its settings and the LCD menu display by sending new menu to the microcontroller. The microcontroller will handle the necessary signals to display the new menu. Implementation wise, the decoding algorithm was based on a look up table to conserve memory space.

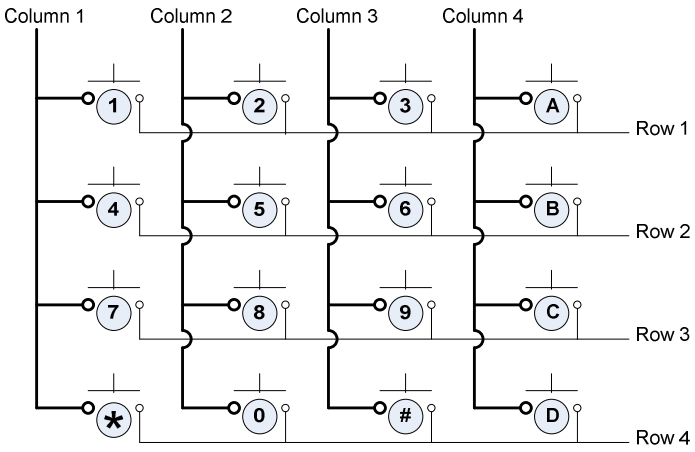


Figure 3.12 : Keypad matrix

3.6 Summary

This chapter described the main hardware modules being used in this research. From the front end, the sensor circuitry was explained and the techniques used to acquire the analogue signals using analogue multiplexers were described. The digital acquisition using SPI for the rectilinear sensors was also presented. Next, the Main Controller Unit being used to control the overall system and analyse the data acquired were discussed. This was followed by details about the User Interface Unit implementation which covered the micro controller HC11, LCD and keypad circuitry.

CHAPTER 4

SOFTWARE IMPLEMENTATION

4.1 Introduction

An optimized system requires that the software architecture to be carefully designed and implemented. Besides that, good programming practices, high performance algorithms and skilled programming techniques enhance the performance of the overall system (Pang, 2004a). In this chapter, the software and firmware implementations for the Data Acquisition, Image Reconstruction Algorithm, Velocity Profile Measurement, Mass Flow Rate Measurement, User Interface Unit and PC GUI Unit are described in detailed.

4.2 Data Acquisition Flow

The data acquisition process is carried out to acquire measurements from the Analogue Signal Conditioning Circuits and the Digital Signal Conditioning Circuit. The digital portion of data acquisition is described first before the analogue portion of data acquisition.

The Digital Signal Conditioning Circuit was triggered to take measurements upon receiving a pulse at the Parallel Load pin of 74HC165 (parallel to serial converter). This will latch the digital values from the digital comparators into 74HC165. By setting the Main Controller Unit as a master device, the SPI system will generate clock signal which was connected to the Shift pin of the 74HC165. This caused the digital data on 74HC165 to be shifted out and read by the Main

Controller Unit at each clock pulse. Since the Main Controller Unit is 16 bit architecture, the SPI was set to read the digital stream in chunks of 16 bits. This is faster because only six reads were needed compared to twelve reads if using 8 bits data. The digital bits were then extracted to respective arrays representing each projection data. These arrays will be used later for image reconstruction and velocity profile measurement.

The next data acquisition process is to take measurements from the Analogue Signal Conditioning Circuit. The channel selection was handled by programming a preset Event Sequencer. The Event Sequencer only needed to be programmed once to take 16 measurements from the specified channels. Once triggered, it will take the measurements according to the channels set in the Event Sequencer. After that, sixteen conversions were executed and the results were stored into arrays in memory. In the DSP, the results were stored left justified in 16 bits registers. Therefore, eight bit right shifts were needed to obtain conversion results in the range of 0 - 255. Next, the 8-bit values acquired from the analogue portion were normalized by using the following equation.

$$Sens[\,i\,] = Vsensor[\,i\,] / 170 \times 128 \qquad \ldots 4.1$$

Where,

Sens[i] = SensNU, SensND, SensMU, SensMD

Vsensor [i] = VsensorND, VsensorNU, VsensorMD, VsensorMU

i = 0..16

The same equation is used for all sensors in the measurement. The maximum voltage of the sensors is set to 2 volts. Since the ADC maximum input voltage is 3 volts, so the maximum digital representation is 170 (2/3 x 255). The other constant (128) is actually 256, but since these data will be divided by 2 later during image reconstruction, it was replaced by the value of 128.

The sixteen conversions would be carried out four times to obtain 64 measurements from the orthogonal projections at Upstream and Downstream layers.

After all conversions were carried out, the program waits for a predefined period (SAMPLING_TIME) in order to fix (make constant) the period per frame. This period will be used later during velocity measurement and will be referred as sampling time (t_s). The whole process was repeated until the number of frames captured equivalent to N. Then, the data analysis was carried out. The data analysis which consists of image reconstruction, velocity profile measurement and mass flow rate measurement will be discussed in the subsequent sections.

The acquisition process block diagram is shown in the following Figure 4.1.

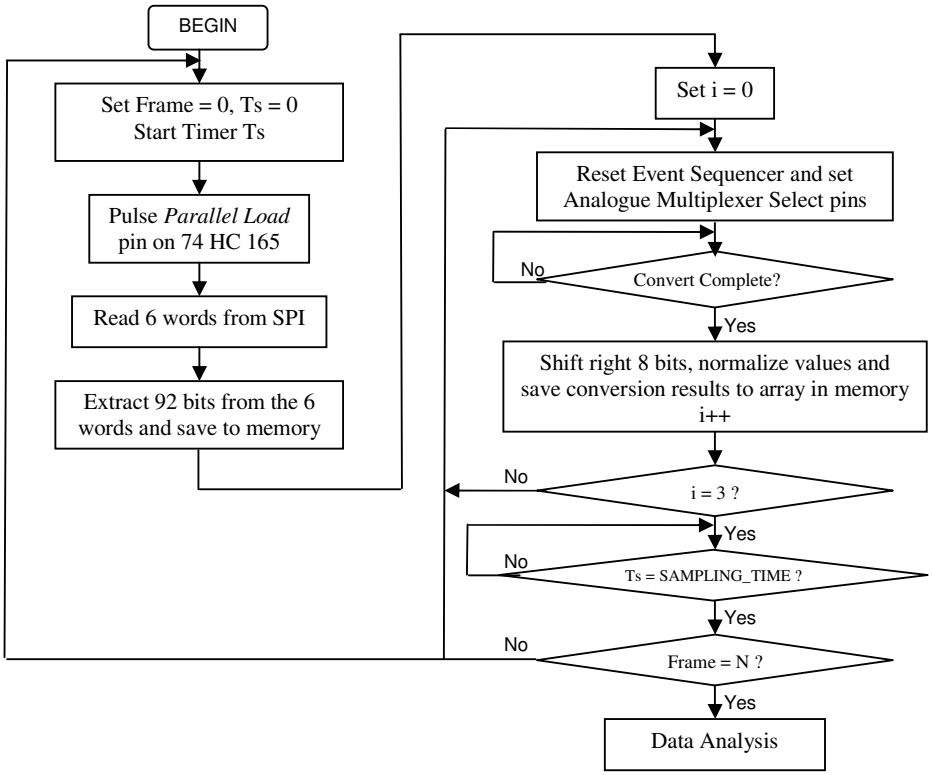


Figure 4.1 : Data acquisition flow chart

4.3 Image Reconstruction Algorithm

In order to visualize the flow concentration using tomography technique, image reconstruction algorithm was applied to the measurement data. The Hybrid reconstruction algorithm which originated from the Linear Back Projection algorithm

was implemented with some improvements. Basically, the back projection algorithm can be illustrated as in the following diagram.

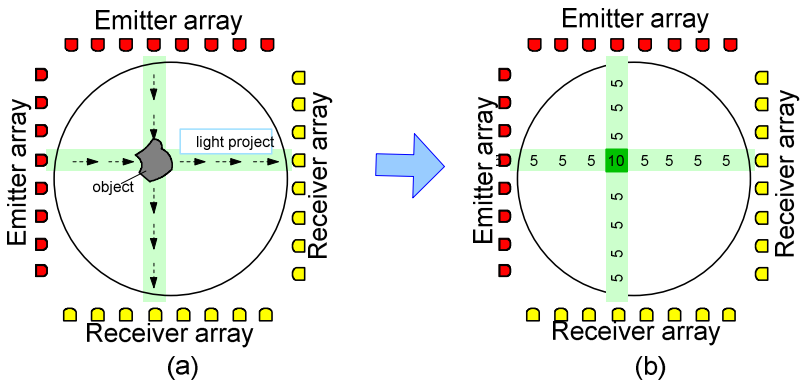


Figure 4.2 : Back projection technique

Figure 4.2 (a) shows the actual object and light projection while Figure 4.2 (b) shows the image reconstructed using Linear Back Projection algorithm. Considering only a single object, this algorithm worked fine (only single pixel detected, shown by dark green) with some smearing, shown by light green. The values shown are the summation of the sensor voltage where sensor voltage is assumed to be 5V if blocked by object. Therefore, the detected pixel shows value of 10V. However, when there is more than one object, the Linear Back Projection algorithm results in ambiguous image as in Figure 4.3. Notice that there are four pixels 'detected' while there are only two actual objects. This is actually a result of summation of the smearing effects.

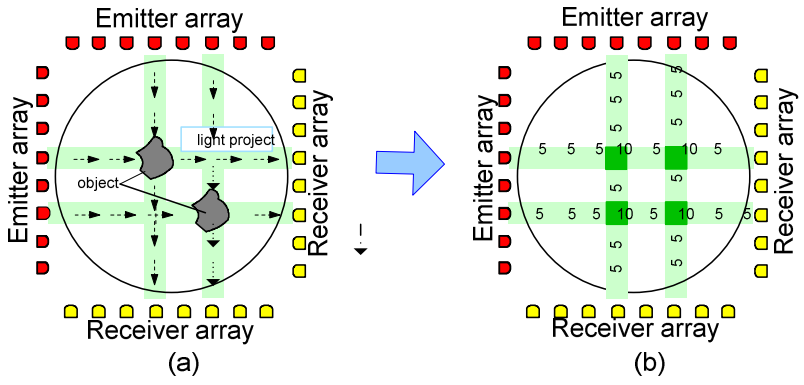


Figure 4.3 : Ambiguous image resulting from back projection

Based on work by Ibrahim *et al.* (1999), ambiguous image can be eliminated by using at least four projections. He added two rectilinear projections to eliminate ambiguous image and named the image reconstruction algorithm as Hybrid Reconstruction algorithm. Pang (2004a) further improved this algorithm by doubling the resolution obtained from the previous research which used 8x8 sensor pairs in the orthogonal layer and 11x11 sensor pairs in the rectilinear layer (Ibrahim *et al.*, 1999) to 16x16 sensor pairs in the orthogonal layer and 23x23 sensor pairs in the rectilinear layer.

Basically, the additional number of sensors used in rectilinear layer was to project the masking layer's beam of light at the center of each pixel, whereby object's presence at any pixel could be detected with only four sensor readings (two from orthogonal and two from rectilinear). In this research, the same hybrid image reconstruction algorithm was improved by further dividing each pixel into eight small triangles. This is shown in Figure 4.4.

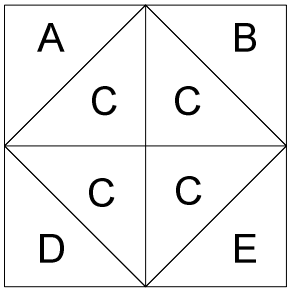


Figure 4.4 : Division of a pixel into eight triangles

Although the pixel was divided into 8 small triangles, four triangles in the center were grouped as one since there wasn't any way to determine the smaller triangles' concentration independently based on the sensor readings. Thus, there were 5 parts, namely A, B, C, D and E. Object that is present at any of these 5 parts could be determined by back projecting the respective sensor readings. By dividing the pixel into 5 smaller parts, each of the small parts can be set as valid or invalid during concentration calculation. This differs from previous work by Pang where the whole pixel was set to a concentration value (using orthogonal layer values) upon confirming that an object exists at that pixel (using only 2 rectilinear layer values).

Using this approach, detection of arc edges of flow object can be presented more accurately. As a fair comparison, let's consider a pixel which is located at the center of the flow region. The total area in this pixel is 28.3 mm^2 for the 85mm diameter pipe being used in this research. Therefore, each triangle contributes one-eight (⅛) of the total pixel, which is 3.5375 mm^2. Using this new Hybrid Reconstruction algorithm, there could be twelve possible cases to represent each pixel instead of only one case. These are shown in Figure 4.5, where triangle parts are shaded to represent blocked location.

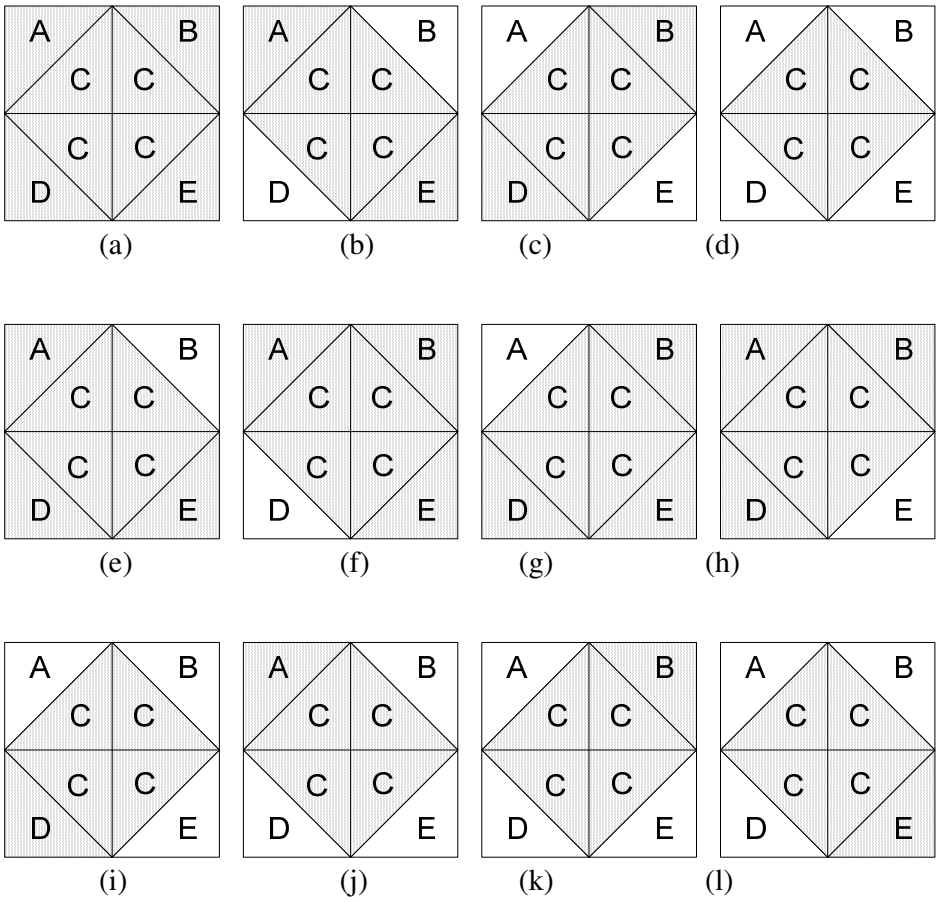


Figure 4.5 : Different cases of pixel reconstruction

From these cases, the percentage of error can be calculated by comparing each case to case (a). This comparison is shown in Table 4.1.

Table 4.1 : Comparison of pixel reconstruction accuracy

Case	Assumption of Object	Area (Hybrid)	Area (new Hybrid)	% Error
a	A cube	28.3000	28.3000	0
b	A diagonally thin object	28.3000	21.2250	25
c	A diagonally thin object	28.3000	21.2250	25
d	A cylinder with small radius	28.3000	14.1500	50
e	An object's arc, top right	28.3000	24.7625	12.5
f	An object's arc, bottom left	28.3000	24.7625	12.5
g	An object's arc, top left	28.3000	24.7625	12.5
h	An object's arc, bottom right	28.3000	24.7625	12.5
i	A small particle with edge	28.3000	17.6875	37.5
h	A small particle with edge	28.3000	17.6875	37.5
k	A small particle with edge	28.3000	17.6875	37.5
l	A small particle with edge	28.3000	17.6875	37.5

Each triangle can be determined as valid or not by considering the following table. OFFSET_P and OFFSET_Q are constants for offset adjustments while iP and iQ are used for array indexing. Equation 4.2 and 4.3 shows the relationship between the array index with x-axis and y-axis.

$$iP = y + OFFSET\_P \quad \ldots 4.2$$

$$iQ = x + OFFSET\_Q \quad \ldots 4.3$$

Where,

y = y-axis location

OFFSET_P = 11

x = x-axis location

OFFSET_Q = -4

VALID_RECT is a constant (0x80) being used to represent the binary bit value of TRUE from the rectilinear sensors acquisition. The value is set to 0 if the bit is FALSE. The following Table 4.2 shows the comparisons made in the software to determine the valid triangles using the New Hybrid Image Reconstruction algorithm. The valid triangle is marked with a tick sign when the comparison equations on the left are fulfilled.

Table 4.2 : Comparison equations to determine valid triangle.

Comparison Equation	Valid Triangle				
	A	B	C	D	E
sensQ[iQ] = VALID_RECT sensP[iP+1] = VALID_RECT	✓				
sensQ[iQ] = VALID_RECT sensP[iP] = VALID_RECT			✓		
sensQ[iQ] = VALID_RECT sensP[iP-1] = VALID_RECT					✓
sensP[iP] = VALID_RECT sensQ[iQ+1] = VALID_RECT		✓			
sensP[iP] = VALID_RECT sensQ[iQ-1] = VALID_RECT				✓	

By using iP and iQ, the number of additions and subtractions carried out per pixel can be reduced compared to calculating it every time the comparison equations are evaluated. Besides that, the comparisons were carried out by evaluating comparison equations that involved iQ first. This prevents the Main Controller Unit from calculating iQ at each comparison. The same method was used for iP. Although it seemed that the reduction was very little, accumulation of the reductions for image reconstruction of more than a hundred sets of data made the reductions significant.

In previous work by Pang (2004a), all the cases shown in Figure 4.5 previously were considered as one case which was case (a). This was because the information from the masking layer was used only to eliminate the ambiguous image and did not contribute to the pixel value (the pixel value was calculated only from the orthogonal layer sensors). By using this New Hybrid Reconstruction algorithm, the information from the masking layer was fully utilized to determine the total value of a pixel.

However, using this new algorithm required at least two bytes; one byte to keep the pixel value and the second byte to carry information on which area was valid (using bit or flag representation). Although this new algorithm brings more accuracy, it affects the overall system performance in a few areas. Firstly, the total data and therefore memory required to store the reconstructed images will be larger.

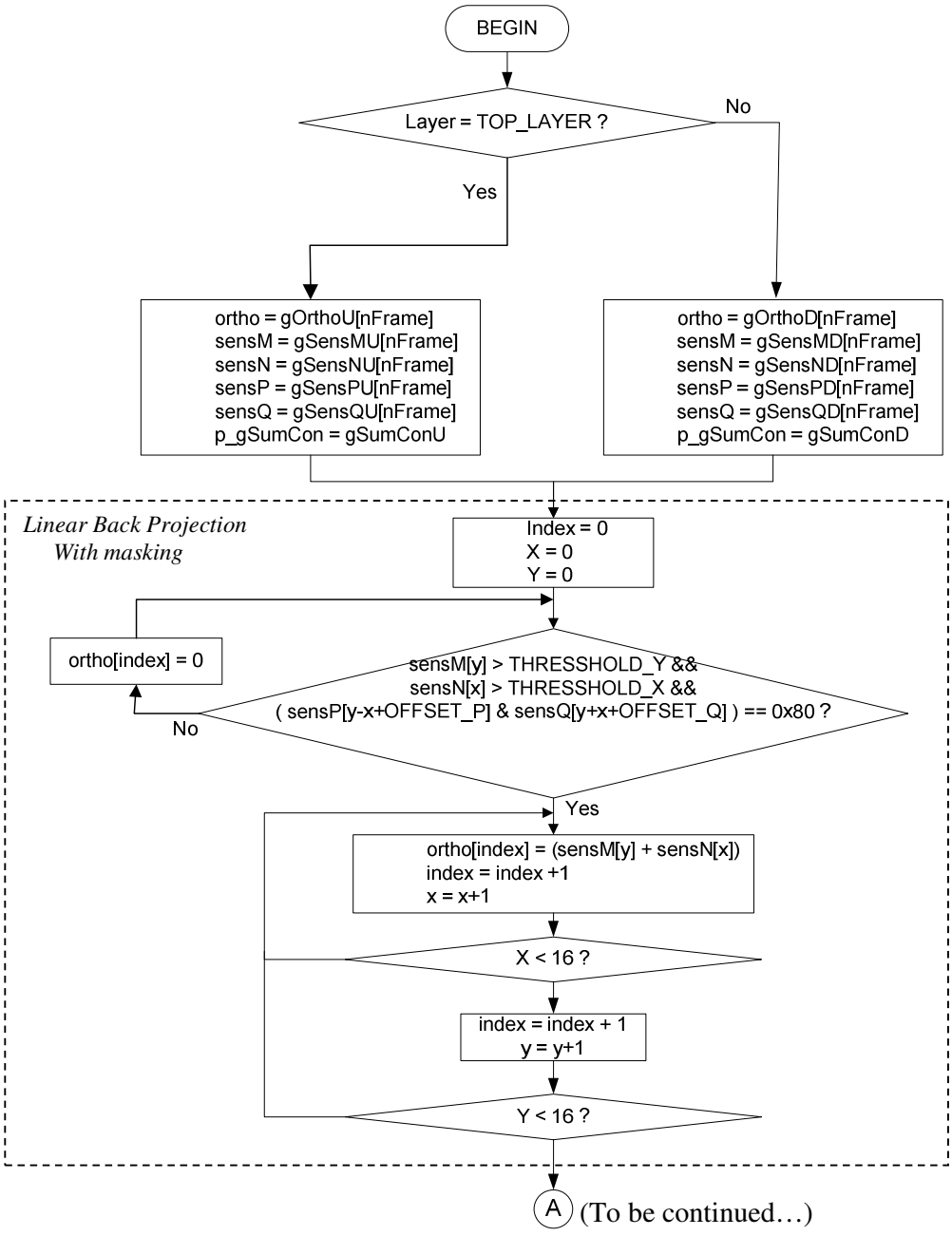
Secondly, the data transfer to the host computer for reconstructed image display will be slower as more data needed to be sent. Thirdly, the amount of cross correlation operation required at later stage to calculate flow velocity will increase and this decreased the system performance.

Therefore, an alternative method was essentially required to represent the equivalent information and yet would not burden the system too much. Finally, this technique was implemented by trading off the actual graphical representation to an equivalent pixel value. This can be understood by realizing that no matter how the pixel concentration was represented by 5 areas; ultimately it would be converted to a single pixel value for mass flow rate measurement. The only difference was that the triangles were displayed as a full pixel with equivalent concentration value of total valid triangles instead of visual representation of actual detected independent triangles. In other words, it was a trade off of the valid triangle location for higher performance but still maintain the more accurate equivalent value of the concentration measurement.

In order to do this, all the triangles were summed up and averaged to represent the whole pixel. This way, the small triangles were taken into account to represent the whole pixel's value. Mathematically, each pixel's value was calculated according to the following equation.

$$V_{Pixel} = \frac{\sum Valid\ Triangles}{8} \quad x \quad Pixel\ concentration \qquad \ldots 4.4$$

The image reconstruction algorithm used in this research is graphically shown in the following flow chart diagram, Figure 4.6.


BEGIN
Layer = TOP_LAYER ?
No
Yes
ortho = gOrthoU[nFrame]
sensM = gSensMU[nFrame]
sensN = gSensNU[nFrame]
sensP = gSensPU[nFrame]
sensQ = gSensQU[nFrame]
p_gSumCon = gSumConU
ortho = gOrthoD[nFrame]
sensM = gSensMD[nFrame]
sensN = gSensND[nFrame]
sensP = gSensPD[nFrame]
sensQ = gSensQD[nFrame]
p_gSumCon = gSumConD
Linear Back Projection
With masking
Index = 0
X = 0
Y = 0
ortho[index] = 0
sensM[y] > THRESSHOLD_Y &&
sensN[x] > THRESSHOLD_X &&
(sensP[y-x+OFFSET_P] & sensQ[y+x+OFFSET_Q]) == 0x80 ?
No
Yes
ortho[index] = (sensM[y] + sensN[x])
index = index +1
x = x+1
X < 16 ?
index = index + 1
y = y+1
Y < 16 ?
A
(To be continued…)

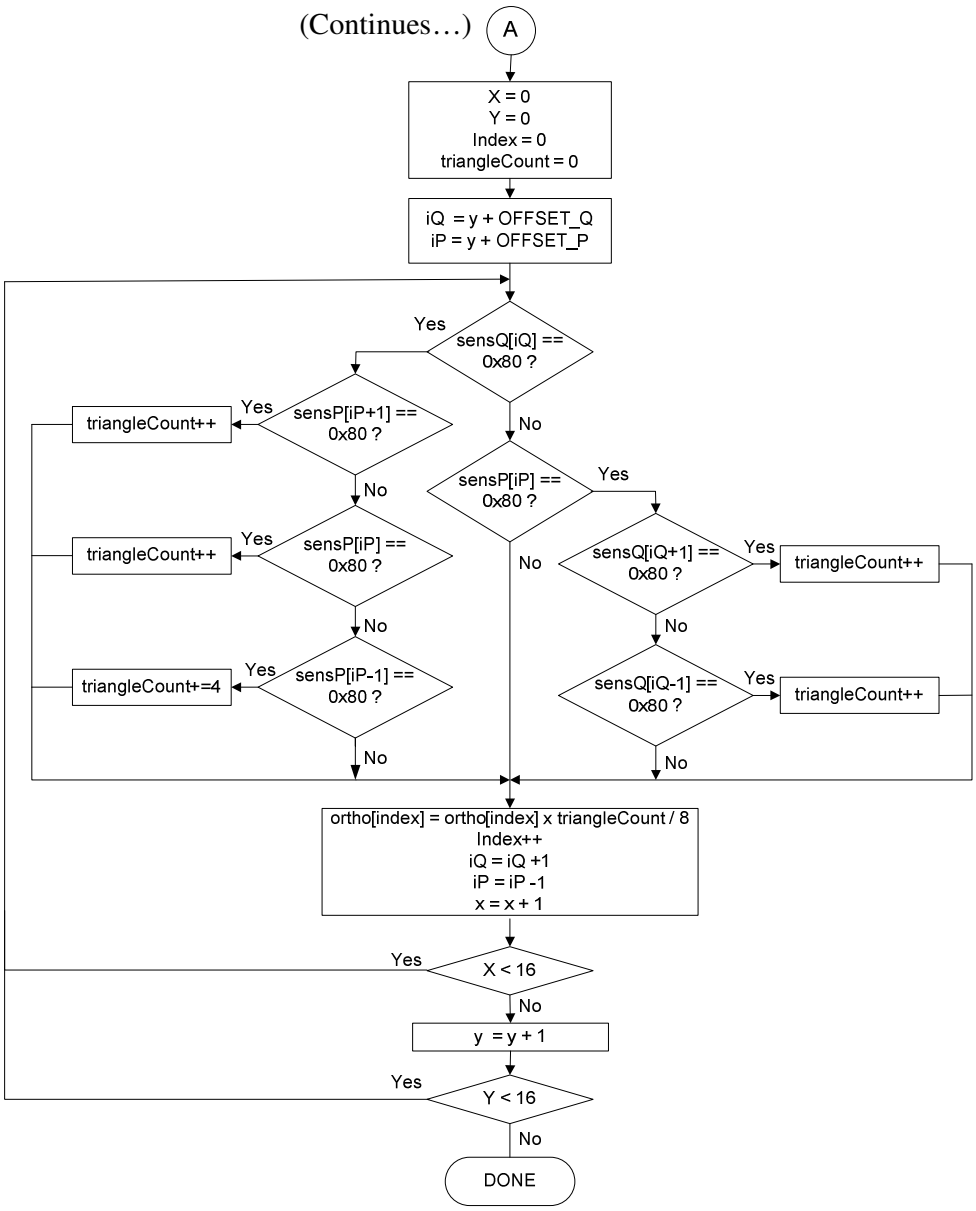


Figure 4.6 : New Hybrid Reconstruction Algorithm

4.4 Velocity Profile Measurement

When a flow passes through a long distance in a gravity flow, it will be affected by velocity, wall friction and collision among the particles (Pang, 2004a). Therefore, not all the particles are travelling at a same velocity. By measuring the flow velocity at each pixel, a profile of flow velocity is obtained and termed as Velocity Profile. The method used to measure the flow velocity at each pixel by cross correlating the upstream pixel values with respective downstream pixel values or upstream sensor values with respective downstream sensor values. The former method of correlation is termed as Pixel to Pixel Correlation (PTP) while the latter is termed as Sensor to Sensor Correlation (STS) (Pang, 2004c). The algorithm to perform cross correlation is explained in the following section.

4.4.1 Cross Correlation Algorithm

As stated in Section 2.4.2, it can be seen that the equation to implement cross correlation function directly requires N^2 operations, where N is the number of data elements. This causes the computation time required to perform cross correlation grows exponentially with the number of data elements. For applications that do not require real time response, this is alright. However, this is not acceptable for online velocity measurement or process control. Pang (2004c) improved this algorithm by means of using data distribution system where the tasks were distributed among a network of high performance computers. In this research, efforts were made to perform cross correlation in acceptable period of time using optimized algorithm in frequency domain and the constraint of embedded system environment.

A detailed understanding on the cross correlation function reveals that it is in fact a time-reversed convolution. This knowledge leads to another fact that convolution can be performed very quickly using a technique known as 'fast convolution' which is used in signal processing field. Fast convolution is actually carried out in frequency or Fourier domain instead of in time domain. Actually, direct implementation in Fourier domain using Discrete Fourier Transform (DFT) also results in the same number of operations (N^2). Cooley and Tukey in 1965 first

described the Fast Fourier Transform (FFT) algorithm which resulted in great reduction of the required number of operations (Dale Grover and John R. Deller, 1999).

The principle of dramatic efficiency results from decomposing the computations into successively smaller DFT computations (Oppenheim *et al.*, 1999). There are two general basic methods for computing the FFT which are the Decimation in Time (DIT) and Decimation in Frequency (DIF). DIT FFT breaks the even-numbered samples into one sequence and the odd-numbered samples into the other sequence. On the other hand, DIF FFT breaks the first N/2 samples for first sequence and the remaining N/2 samples for the other sequence. The difference of the two methods tends to be in the order of the input and output. DIT FFT expects the input sequence to be bit reversed order and outputs a normal order sequence while DIF FFT expects a normal order sequence and outputs bit reversed order sequence.

There are many different versions of FFT algorithm being optimized for use in different conditions. Normally, decomposition of the DFT into two parts at each pass requires that the length of the sequence to be a power of 2. This is known as the radix-2 FFT algorithm. There are also versions of FFT used for odd length sequences, even length sequences and prime number length sequences. Besides radix-2, FFT can also be implemented with radix 4 or radix 8. In some cases, the sequence is simply zero padded until the number of elements reaches a power of 2 and suitable for Radix-2 FFT implementation.

FFT is used to convert time domain sequences to frequency domain while Inverse Fast Fourier Transform (IFFT) is used to convert the frequency domain sequences back to time domain. Actually, FFT and IFFT algorithm is almost similar but not the same. This can be noted by comparing Equation 4.5 (FFT) and Equation 4.6 (IFFT).

$$X[k] = \sum_{n=0}^{N-1} x[n](W_N)^{kn} \qquad \ldots 4.5$$

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k](W_N)^{-kn} \qquad \ldots 4.6$$

Where,

X[k] = Frequency domain sequence

k = Index in frequency samples

n = Index in time samples

W_N = Twiddle factor

N = Number of elements

Equations 4.6 differ from Equation 4.5 at the power of the twiddle factor which is negative and also the overall sum is divided by N in Equation 4.5. Practically, the algorithm is developed with flexibility to either perform FFT or IFFT by setting a flag which will take care of the differences in the equations. This saves code space of the embedded system. In the equations, W_N which is the twiddle factor is given by the following equation.

$$(W_N)^{kn} = \left(\cos\left(\frac{2\pi}{N}\right) - j\sin\left(\frac{2\pi}{N}\right)\right)^{kn} \qquad \ldots 4.7$$

This equation can also be written as Equation 4.8 or Equation 4.9.

$$(W_N)^{kn} = \cos\left(\frac{2\pi kn}{N}\right) - j\sin\left(\frac{2\pi kn}{N}\right) \qquad \ldots 4.8$$

$$(W_N)^{kn} = e^{-j\frac{2\pi kn}{N}} \qquad \ldots 4.9$$

For implementation in embedded system, it is more suitable to use Equation 4.7. The values for all the twiddle factors are pre-calculated and stored in the memory for use during FFT or IFFT calculations. This will prevent from re-calculating these values for each FFT or IFFT. The following Figure 4.7 shows a typical example of 8 point Radix-2, DIT FFT implementation.

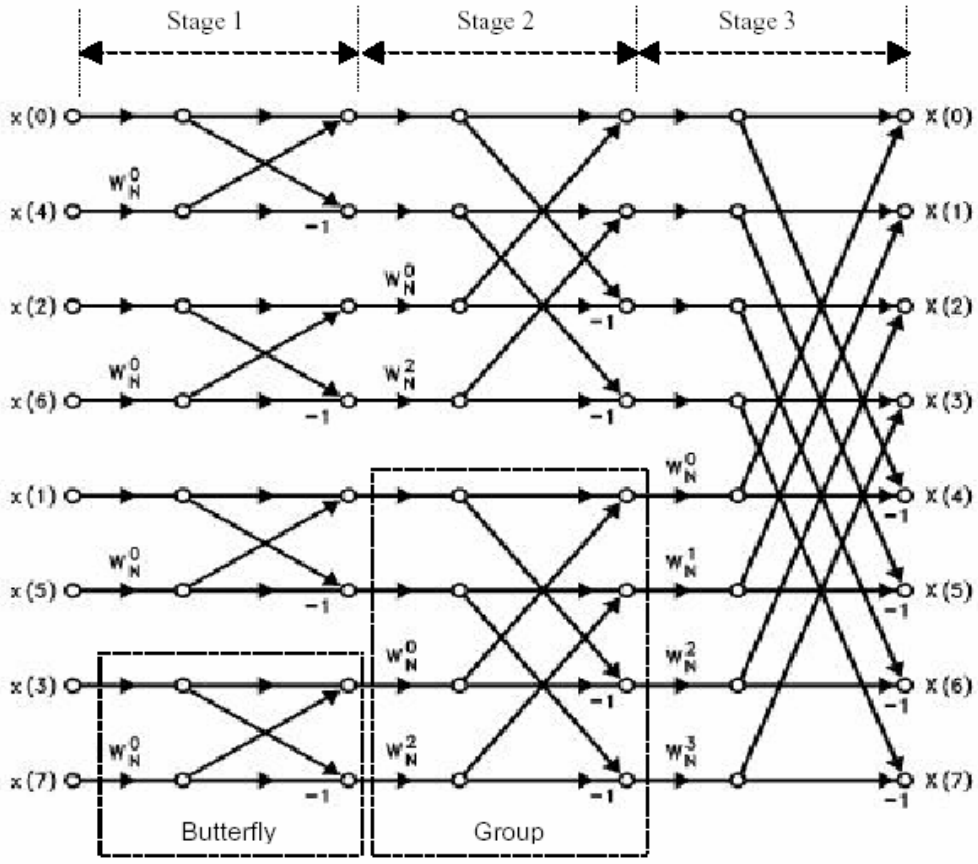


Figure 4.7 : Radix-2, DIT FFT algorithm

First of all, FFT algorithm requires that the input data to be in bit reversed order. The term bit reverse can be explained using Table 4.3.

Table 4.3 : Bit reverse indexing

Original index	**Original (Binary)**	**Bit Reversed (Binary)**	**Bit Reversed index**
0	000	000	0
1	001	100	4
2	010	010	2
3	011	110	6
4	100	001	1
5	101	101	5
6	110	011	3
7	111	111	7

Considering a data sequence with 8 elements (N = 8) as shown in Table 4.5, three binary bits are required to sufficiently represent the index. The Bit Reversed Binary is obtained by reading the Original (Binary) from right to left, bitwise. The resulting Bit Reversed Index will be used as pointers to extract data from the external data memory locations to internal data memory locations. This made sure that the input array for the FFT is in bit reversed order.

In order to save processing time, the Bit Reverse Index was generated once during program initialization and used through out the program. The algorithm to generate the bit reverse index is shown in the following Figure 4.8. This algorithm works by checking the bits from most right bitwise and 'mirror' the bit to the opposite side if the bit is 1. This 'mirroring' is equivalent to reading the bits in reverse order.

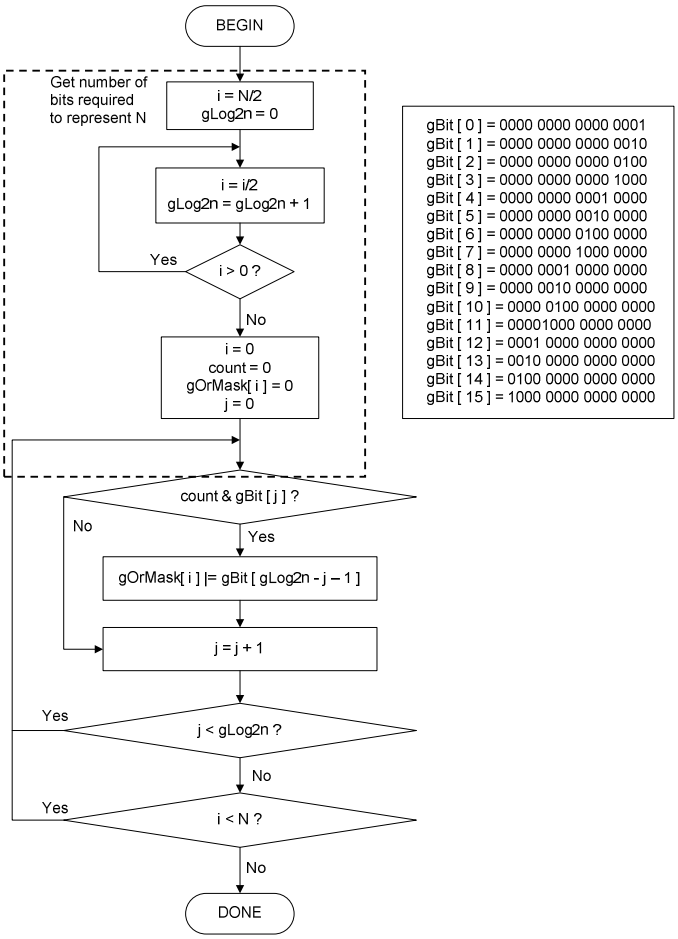


Figure 4.8 : Algorithm to generate bit reverse index

For N elements of data for convolution, direct convolution function in time domain or direct DFT implementation requires N^2 operations while DIT FFT (radix 2) only requires $2N\log_2 N$ operations (1 FFT and 1 IFFT). As the number of data elements, N increases, savings of operations are more significant. The following table shows the comparison of convolution function performed in time domain and frequency domain in terms of number of data elements and its required number of operations. It can be noted that the percentage of speed increment of using Frequency Domain is from 33.33% (which is about 0.3 times faster) up to 9209.09% (which is about 92 times faster) than similar operation in time domain.

Table 4.4 : Speed comparison between convolution in time and in frequency domain.

Data Elements, N	**Number of Operations required**		**% of Speed Increase**
	Time Domain	**Frequency Domain**	
8	64	48	33.33%
16	256	128	100.00%
32	1,024	320	220.00%
64	4,096	768	433.33%
128	16,384	1792	814.29%
256	65,536	4096	1500.00%
512	262,144	9216	2744.44%
1024	1,048,576	20480	5020.00%
2048	4,194,304	45056	9209.09%

In performing cross correlation to determine time delay, the values of interest are the correlation coefficients in $r_{xy}(n)$. These coefficients allow time delay to be determined by locating the element where the coefficient is at the peak value. The peak value shows that the correlation is the highest or in other words, the two signals being correlated is most similar at this time delay. In order to complete the cross correlation, IFFT is needed to convert the result from Fourier domain to time domain. Therefore, the cross correlation function can be implemented as in the following block diagram shown in Figure 4.8. The lag can then be converted to time by multiplying it with sampling time, t_s.

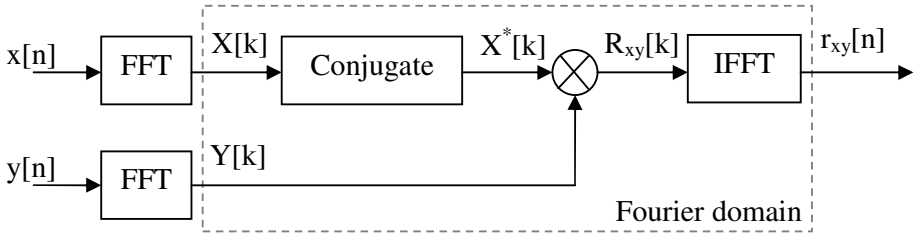


Figure 4.9 : Cross correlation implementation in Fourier domain

Mathematically, the cross correlation in Fourier domain can be expressed as in Equation 4.10.

$$r[n] = \text{IFFT} \{ \text{FFT}(x[n]) \times \text{FFT}(y[n])^* \} \ldots 4.10$$

Where,

r[n] = Cross correlation coefficients

x[n] = First sequence

y[n] = Second sequence

* = Conjugate operator

4.4.2 Optimized Cross Correlation in Fourier Domain

In this research, optimizations were carried out to perform cross correlation rapidly. As explained in the previous section, cross correlation was performed in Fourier domain in order to reduce the number of operations required. However, the FFT algorithm expects complex values input while the sensors values are in real values. Commonly, the real values are converted to complex values by zeroing the imaginary part. This conversion does not fully utilize the FFT algorithm.

Further reduction of operations was proposed here by taking advantage of real values sequences acquired from the sensors values. Actually, the operations carried out on these values by the FFT algorithms are really equivalent to performing twice of half FFTs. Therefore, instead of performing two separate FFTs, only a

single FFT with some additional steps are sufficient to transform two real values sequences. This method will hereafter be referred as RFFT and its inverse as IRFFT. Basically, the RFFT method is carried out by arranging the first sequence as the real values and the second sequence as the complex values. Then, the FFT is carried out as normal. The resulting complex values can be separated to two sequences by a simple pass.

The optimized cross correlation function implementation using RFFT and IRFFT in Fourier domain is shown in the following block diagram.

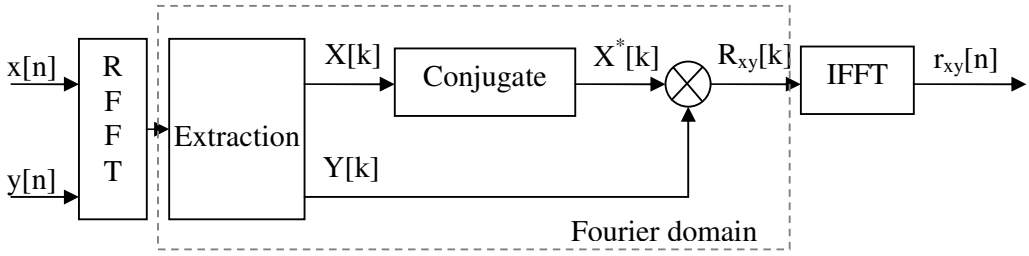


Figure 4.10 : Optimized cross correlation implementation in Fourier domain

Figure 4.9 differs from Figure 4.8 at the front end, where two separate FFTs were replaced by a single RFFT followed by an Extraction module. The Extraction module is required to extract the combined output (complex values) into two separate sequences (real values). Standard FFT method requires $2N\log_2N$ operations ($N\log_2N$ operations per FFT) while RFFT method only requires $N(1+\log_2N)$ operations ($N\log_2N$ for FFT and N for Extraction). Therefore, the ratio is $(2\log_2N) / (1+ \log_2N)$. The following table shows the percentage of speed increase for comparison of different lengths of N for the front end implementation using FFT and RFFT.

For this research, the length being used is 128 which entitled for improvement of 75 % (which is about 0.75 times faster). Therefore, the cross correlation algorithm should be performed in Fourier Domain instead of Time Domain. Besides that, the cross correlation should also implement RFFT because it is faster compared to FFT.

Table 4.4 : Speed comparison of FFT method and RFFT method.

Sequence Length, N	Number of Operations Required		% of Speed increase
	FFT	RFFT	
8	48	32	50.00%
16	128	80	60.00%
32	320	192	66.67%
64	768	448	71.43%
128	1792	1024	75.00%
256	4096	2304	77.78%
512	9216	5120	80.00%
1024	20480	11264	81.82%
2048	45056	24576	83.33%

4.4.3 Pixel to Pixel Correlation

In order to perform Pixel to Pixel Correlation, a large number of data frames need to be acquired and stored in memory. These values were output from the image reconstruction algorithm where the data frames or images reconstructed were stored frame by frame in the external memory. For each pixel to pixel correlation, the pixels values were extracted from the data frames and stored in DSP internal memory for faster data access. This was done so since many operations would be performed on these values and therefore the memory access time should not constraint the performance. The internal memory allows read and write operations to be carried out with zero wait state compared to external memory which requires at least one wait state. Therefore, some arrays were created temporarily at each correlation to be used as fast calculation space. Each element in the array is actually a pixel value extracted from each frame. This technique is illustrated in Figure 4.11.

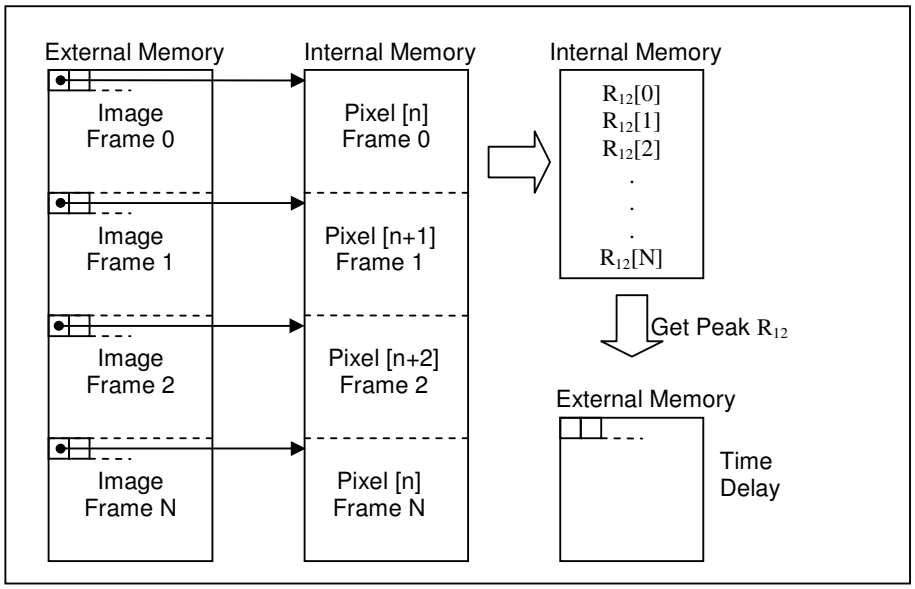


Figure 4.11 : Illustration of correlation performed in the internal memory

4.4.4 Sensor to Sensor Correlation

For sensor to sensor correlation, the upstream sensors values were cross correlated to the downstream sensors values. These values were acquired during the data acquisition process where the acquired values were stored frame by frame in the external memory. The only difference is that the values were stored according to sensors (sensMU, sensNU, sensMD, sensND). The same principle as in Pixel to Pixel Correlation was implemented here, where the sensor values were extracted from the external memory into internal memory as required in each sensor correlation for faster processing.

Using Sensor to Sensor Correlation, the number of cross correlations required was reduced to only the number of sensors per sensor layer. This resulted in significant reduction of the operations required. If we consider 16x16 parallel projection sensors as used in this research, the total cross correlations required for velocity profile measurement were only 32 correlations compared to 256 correlations if using Pixel to Pixel Correlation. This significant reduction certainly motivates attempt to use this approach in obtaining flow velocity profile.

As mentioned previously in Section 2.4.2, Sensor to Sensor correlation does not result in velocity profile. Therefore, efforts were attempted in this research to generate the velocity profile by back projecting and averaging the velocity acquired. In fact, this method is a reversed order of what had been done with Pixel to Pixel Correlation where back projection was carried out first in the image reconstruction algorithm before cross correlation took place. However, there was a significant reduction of operations required compared to the former method.

Implementation wise, the pixel to pixel cross correlation algorithm and sensor to sensor cross correlation algorithm were integrated within one same function. This was done so due to the fact that both algorithms are actually the same. Only the memory locations were different. Global variable, *gVel* was used to determine which cross correlation to perform and point the data array to the appropriate memory locations. The function will extract the correct values from the correct memory locations during execution for cross correlation.

4.5 Mass Flow Rate Measurement

This research was carried out with the ultimate goal of performing mass flow rate measurement of solid flow. The image reconstruction algorithm and velocity profile generation algorithm were implemented to facilitate this purpose. Basically, the mass flow rate can be obtained according to Equation 2.1 which requires flow image concentration, flow velocity profile and a constant which is the object's density.

Since the algorithm required concentration area, the image reconstructed from sensor voltages cannot be used directly. These image values were converted to area unit by normalizing the values and multiplying them to respective weighted ratio constants as in Equation 4.11.

$$Area(x,y) = V_{pixel} \times W_A(x,y) \quad \ldots 4.11$$

Where,

$Area(x,y)$ = Area at pixel (x,y)

V_{pixel} = Normalized Sensor Value

W_A = Pixel Weighted Ratio at pixel (x,y)

W_A was calculated by summing the total area of each pixel in the tomogram and the values are shown in Table 4.6. These weighted ratio constants were pre-calculated and stored in the DSP memory. As can be seen from this table, the full pixel carries a weighted sum of 2830. The total value of all pixels summed up is 567,452. This is actually a fixed point representation of 5674.52 mm^2 which is the total area of the whole tomogram in mm^2 (obtained from πr^2, where r is tomogram radius = 42.5 mm). The reason for using this representation was to prevent from the use of floating point data representation which required longer processing time on a fixed point processor.

Table 4.6 : Weighted ratios of the tomogram image area

0	0	0	0	474	1642	2403	2773	2773	2403	1642	474	0	0	0	0
0	0	74	1669	2820	2830	2830	2830	2830	2830	2830	2820	1669	74	0	0
0	74	2123	2830	2830	2830	2830	2830	2830	2830	2830	2830	2830	2123	74	0
0	1669	2830	2830	2830	2830	2830	2830	2830	2830	2830	2830	2830	2830	1669	0
474	2820	2830	2830	2830	2830	2830	2830	2830	2830	2830	2830	2830	2830	2820	474
1642	2830	2830	2830	2830	2830	2830	2830	2830	2830	2830	2830	2830	2830	2830	1642
2403	2830	2830	2830	2830	2830	2830	2830	2830	2830	2830	2830	2830	2830	2830	2403
2773	2830	2830	2830	2830	2830	2830	2830	2830	2830	2830	2830	2830	2830	2830	2773
2773	2830	2830	2830	2830	2830	2830	2830	2830	2830	2830	2830	2830	2830	2830	2773
2403	2830	2830	2830	2830	2830	2830	2830	2830	2830	2830	2830	2830	2830	2830	2403
1642	2830	2830	2830	2830	2830	2830	2830	2830	2830	2830	2830	2830	2830	2830	1642
474	2820	2830	2830	2830	2830	2830	2830	2830	2830	2830	2830	2830	2830	2820	474
0	1669	2830	2830	2830	2830	2830	2830	2830	2830	2830	2830	2830	2830	1669	0
0	74	2123	2830	2830	2830	2830	2830	2830	2830	2830	2830	2830	2123	74	0
0	0	74	1669	2820	2830	2830	2830	2830	2830	2830	2820	1669	74	0	0
0	0	0	0	474	1642	2403	2773	2773	2403	1642	474	0	0	0	0

Implementation wise, the mass flow rate algorithm were not calculated sequentially as in the equations presented. This was due to the limitation of fixed point processing on the Main Controller Unit. Basically, the difference is that multiplications and divisions were performed alternately, where multiplication is carried out first and then followed by division. Performing all multiplications first will cause the data to overflow (exceeds long integer data range) while performing

divisions first will cause the data to be less precise (due to round up to integers). Hence, alternating between the two operations prevent from the data range problem. The overall algorithm for calculating the Mass Flow Rate is shown in the following figure.

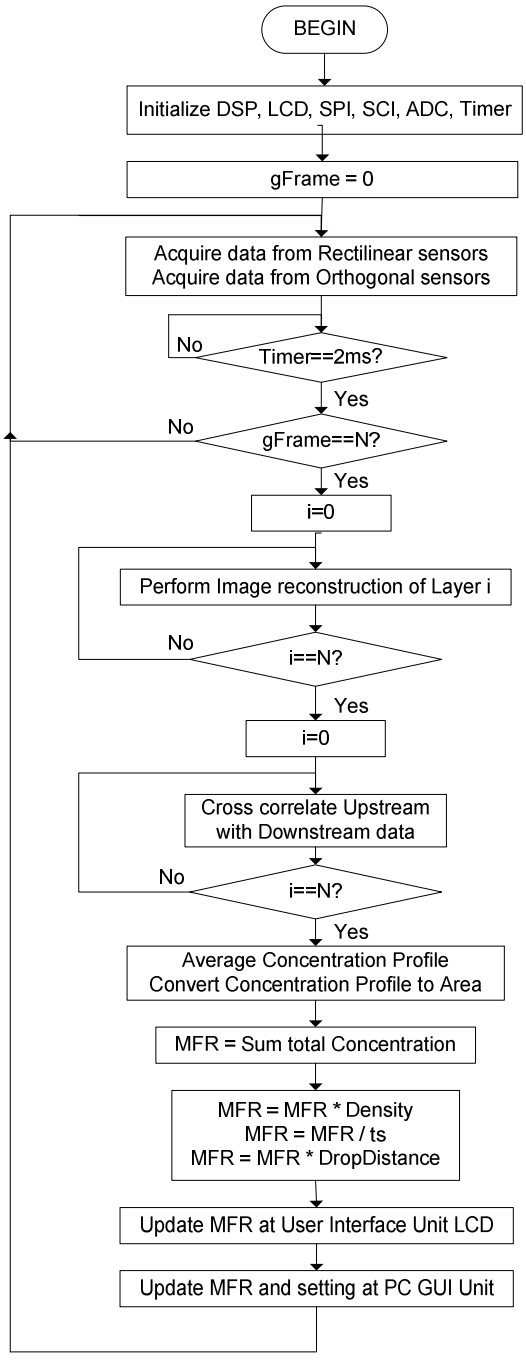


Figure 4.12 : Overall algorithm for calculating Mass Flow Rate

4.6 User Interface Unit

In this section, two areas of discussion will be carried out since these are the main tasks performed by this unit. The first part is the LCD communication protocol with the Main Controller Unit while the second part is the keypad decoding algorithm. Following that, the overall operation or program flow of User Interface Unit will be shown.

4.6.1 LCD Communication Protocol

The communication protocol between the Main Controller Unit and the User IO Unit is described in this part. The Main Controller Unit acts as a master device, where most actions are controlled by it. A set of commands was established so that the User IO Unit can understand the Main Controller Unit's instructions. These commands set were mainly derived from the LCD module commands set and are listed in the following table. All communications use 2 bytes transfer. The first byte is used to determine whether the second byte is an instruction or a data. This allows the necessary bit (RS) at the LCD module to be set to logic HIGH or LOW.

Table 4.7 : Commands set for communication between Main Controller Unit and User IO Unit

Command	Description
LCD_DISPLAY	Turn display ON or OFF
LCD_CLEAR	Clears the display
LCD_LINE1	Set memory pointer to line 1 or line 2
MFR_UPDATE	Set memory pointer to MFR result location

During start up, the Main Controller Unit initializes the User IO Unit by sending menu display characters to it. This will interrupt the User IO Unit to receive the characters and display it onto the LCD module. After the display routine completes, the User IO Unit continues to decode keypad to determine if any key is pressed. This is repeated until a key press triggers the User IO Unit to send the value of the pressed key to the Main Controller Unit. If the value is a valid menu

command, the Main Controller Unit will send updated menu characters to the User IO Unit for display update. Otherwise, the value will be ignored and the whole process is repeated.

For the menu display, the character strings were stored in the Main Controller Unit so that changes to the menu string can be made without performing modifications to the User IO Unit. In general, the User IO Unit only sends in keypad pressed as inputs to the Main Controller Unit and receives commands or characters from Main Control Unit to display on LCD module. The Main Controller Unit checks during intervals on whether any key is pressed by checking if any character arrives to the serial buffer. In the event that a key being pressed, the receive buffer will contain the key value and the Main Controller Unit will wait for another serial data that contains the option that user chooses.

For result display, the Main Controller Unit was programmed to send updated Mass Flow Rate results to the User IO Unit upon completion of each analysis. Since the location for the result characters is fixed, the Main Controller Unit only updates that particular location using the command set MFR_UPDATE. This saves time as the other string characters need not be updated.

4.6.2 Keypad Decoding Algorithm

As presented in Section 3.5.3, the key press on the key pad was decoded using standard software based technique. The algorithm to detect the key press is shown in Figure 4.13.

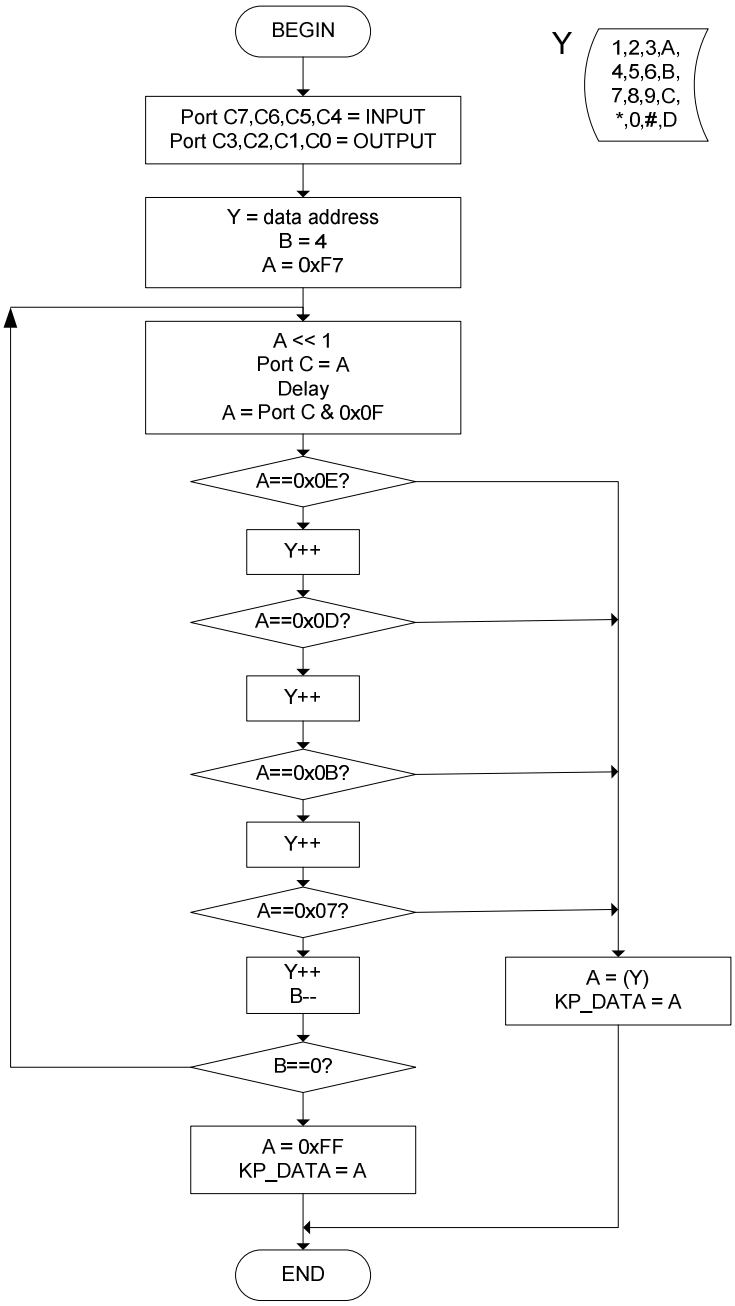


Figure 4.13 : Keypad decoding algorithm

4.6.3 Overall Operation of User Interface Unit

The overall operation of the User Interface Unit is shown by the flow chart in Figure 4.14. The micro controller (HC11) begins execution upon power up and initializes the GPIO, LCD, interrupt vector and Serial Communication settings. Then,

it starts the keypad decoding as described in previously shown Figure 4.13. At any point of time, the Main Controller Unit may send command to change the LCD settings or data to be shown on the LCD. HC11 will be triggered to execute the Interrupt Service Routine (ISR) for Serial Communication (Character Received). In the ISR, HC11 will obtain two characters, the former to determine the latter is an instruction or data. Based on the first character, HC11 will execute the appropriate subroutine. Upon completion, HC11 will return from interrupt.

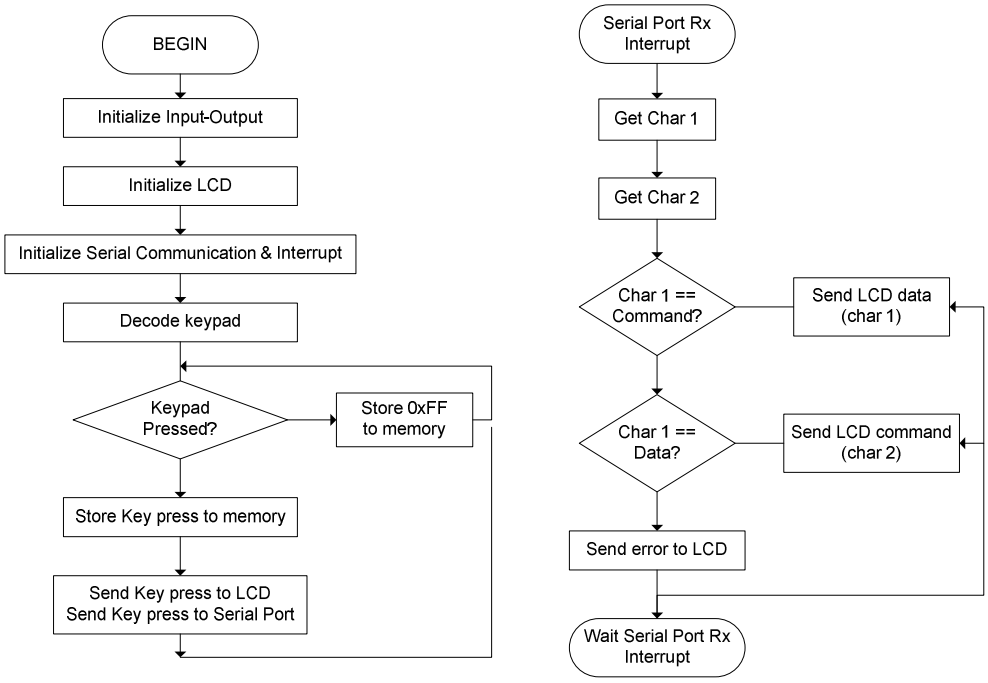


Figure 4.14 : Overall operation of User Interface Unit

4.7 PC GUI Unit

PC GUI Unit is a software module developed on a personal computer platform. This module was developed in order to visualize and verify results obtained from the Main Controller Unit. Besides, this module was also useful in term of assisting trouble shooting. Basically, this unit has several functions such as displaying tomogram images reconstructed by the Main Controller Unit, displaying sensor voltages in a graph and changing certain settings on the Main Controller Unit. Shown in Figure 4.15 is a capture of running PC GUI Unit.

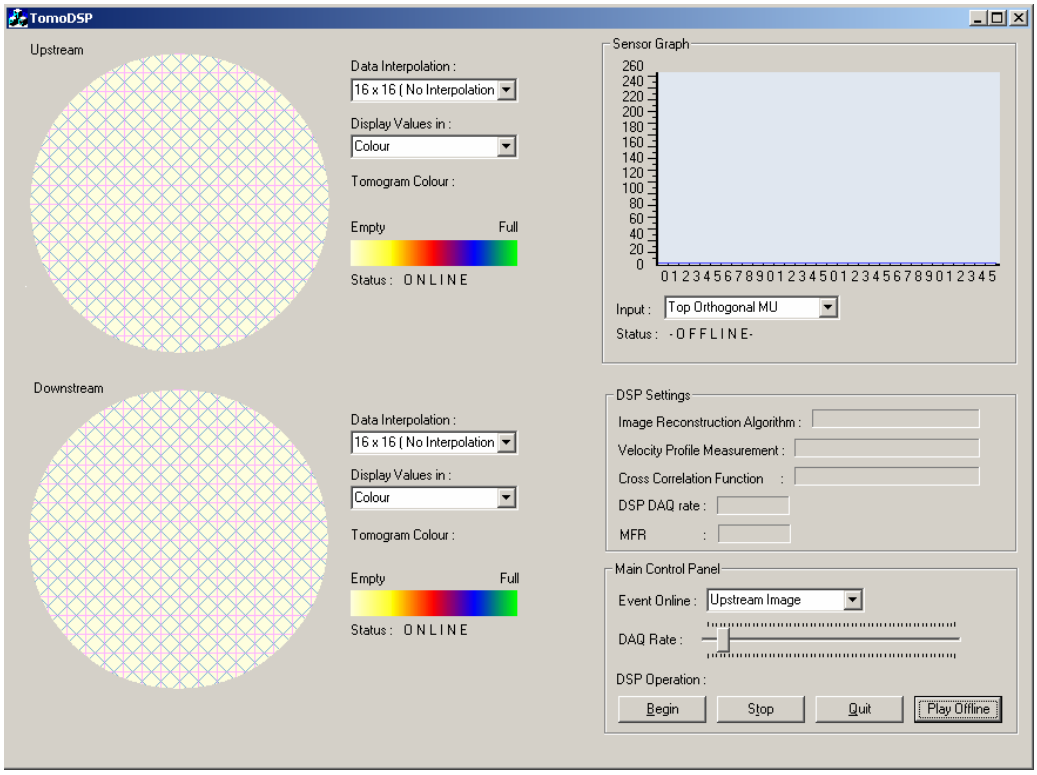


Figure 4.15 : PC GUI Unit capture

4.7.1 Tomogram Display

The tomogram image display is essential in the sense that it visualizes the data acquired and the results obtained from analysis. Different flow pattern can also be seen visually from the tomogram images. Since there are two tomogram display (one for Upstream layer, one for Downstream layer), flow occurring at the respective layers can be determined. This is important in verifying the data acquisition and image reconstruction algorithm. Voltage levels are represented by different colour range chosen by user. By default, white colour represents zero voltage while red colour represents full voltage. The colour bar is shown by Figure 4.16 while an example tomogram is shown in Figure 4.17.

Figure 4.16 : Intensity colour bar

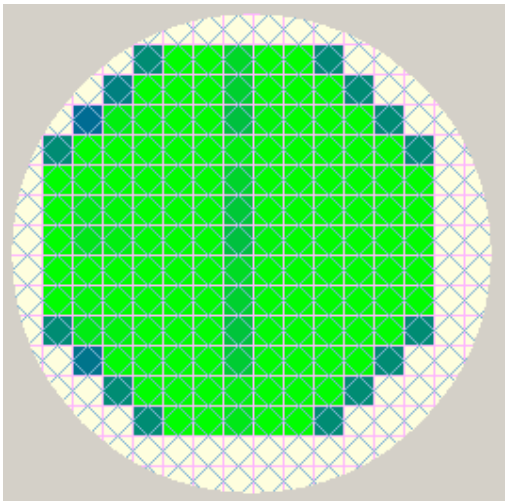


Figure 4.17 : Example of image reconstruction tomogram

4.7.2 Sensor Graph Display

The sensor graph Display is used to display the current values being acquired by the Main Controller Unit using a bar graph. This is very useful during trouble shooting to verify that the sensor data is correct. Each projection's sensor values can be selected to be displayed on the graph. There are two settings for the graph display which are the setting for orthogonal projection sensors and setting for rectilinear projection sensors. For the former setting, there are 16 sensor values while the latter setting displays 23 sensor values. Besides that, there are also selections of Upstream Sensors or Downstream Sensors. All together, there are 8 choices, which are Upstream sensMU, sensNU, sensPU, sensQU or Downstream sensMD, sensND, sensPD, sensQD. This graph is disabled during tomogram image display to reduce serial communication transfer. An example of Sensor Graph Display was captured and shown in Figure 4.18.

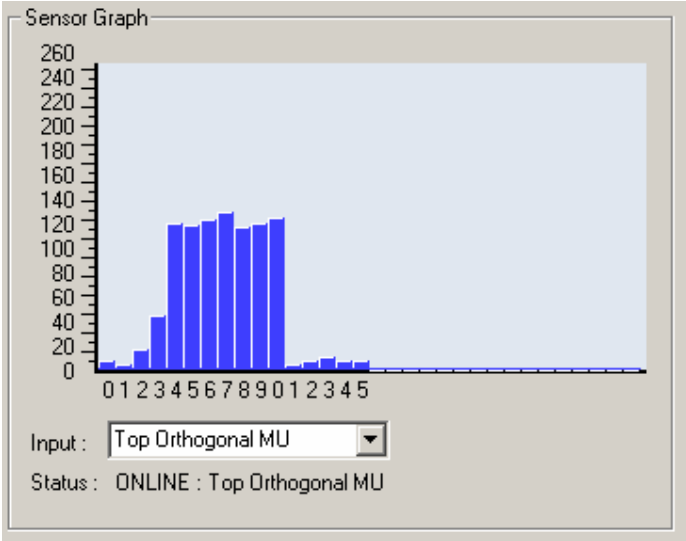


Figure 4.18 : Example of sensor graph display

4.7.3 PC GUI Communication Protocol

Communication protocol was developed to synchronize communication between the PC GUI unit and the Main Controller Unit. Basically, it was a message based communication where the host (PC GUI) sends a command and the Main Controller Unit performs some actions or responses with data. The commands available are listed in Table 4.8.

Table 4.8 : Commands for PC GUI and Main Controller Unit communication

Command	Value	Function
BOTTOM_LAYER	0x0001	Acquire bottom layer
TOP_LAYER	0x0002	Acquire top layer
BOTH_LAYERS	0x0003	Acquire both layers
ORTH_MD	0x0004	Acquire sensor array for projection MD
ORTH_ND	0x0005	Acquire sensor array for projection ND
ORTH_MU	0x0006	Acquire sensor array for projection MU
ORTH_NU	0x0007	Acquire sensor array for projection NU
RECT_QU	0x0008	Acquire sensor array for projection QU
RECT_PU	0x0009	Acquire sensor array for projection PU
RECT_QD	0x000A	Acquire sensor array for projection QD
RECT_PD	0x000B	Acquire sensor array for projection PD
VELOCITY	0x000C	Acquire velocity profile
MASSFLOWRATE	0x000D	Acquire mass flow rate only

(To be continued…)

(Continue…)

NEW_ALGO	0x0010	Use new improved masking algorithm
OLD_ALGO	0x0020	Use linear back projection algorithm
OLD_ALGO_MASK	0x0030	Use back projection algorithm with masking
TIME_CCF	0x0040	Cross correlate in time domain
FREQ_CCF	0x0050	Cross correlate in frequency / Fourier domain
VEL_PIXEL	0x0060	Perform Pixel to Pixel correlation
VEL_SENSOR	0x0070	Perform Sensor to Sensor correlation

For both the Main Controller Unit and the PC GUI unit, the number of bytes for transfer was set according to the command sent and received. For example, tomogram display functions such as BOTTOM_LAYER and TOP_LAYER requires 256 bytes of data to be transferred while sensor values display function such as ORTH_MD, ORTH_ND only requires 16 bytes data. Certain messages were only sent to change the settings for image reconstruction or velocity profile measurement.

4.8 Summary

This chapter discussed thoroughly the software implementation and optimization techniques used in this research. The Data Acquisition Flow which was first presented touched on the analogue acquisition using high speed ADC and digital acquisition using combination of voltage comparators and parallel to serial converter. The proposed methodology was to reduce the number of analogue to digital conversions required by the system and thereby increase the system's performance. Next, a new image reconstruction algorithm improvement was proposed which is the New Hybrid Reconstruction algorithm. This algorithm improved the image reconstruction accuracy while maintaining the performance.

The discussion was continued with the velocity profile measurement where the cross correlation algorithm was explained in detail. Optimizations using Fourier domain and RFFT were suggested to reduce the processing time and increase the overall performance. Back projecting the results obtained from Sensor to Sensor correlation in order to generate velocity profile was also suggested. Following that, the overall algorithm used in mass flow rate measurement of this system was presented.

The working algorithms for User Interface Unit were presented with focus on the LCD communication protocol with the Main Controller Unit and the keypad decoding algorithm. Finally, the details of PC GUI Unit were described with emphasis on the tomogram display, sensor graph display and communication protocol with the Main Controller Unit.

CHAPTER 5

RESULTS

In this chapter, the results obtained from experiments and studies carried out during the research are presented and discussed. Generally, the results were grouped into four parts, which are the data acquisition system, image reconstruction, velocity profile measurements and mass flow rate measurements. The performance of the tasks being carried out during experiments were also logged and discussed.

5.1 Data Acquisition System

Data Acquisition System in this research comprised of the Signal Conditioning Unit together with the Main Controller Unit. This system is responsible to condition incoming sensor voltages and acquire measurements from the Sensor Unit. Basically, the Signal Conditioning Unit comprises of two parts, the Analogue Signal Conditioning and the Digital Signal Conditioning. One Printed Circuit Board (PCB) was fabricated for each part per layer. All together, there were four PCBs fabricated for Signal Conditioning Unit. Figure 5.1(a) shows the Single Layer of Analogue Signal Conditioning PCB, Figure 5.1(b) shows the Analogue Signal Conditioning PCB in 2 layers, Figure 5.1(c) shows Single Layer of Digital Signal Conditioning PCB and Figure 5.1(d) shows Digital Signal Conditioning PCB in 2 layers. Figure 5.2(a) shows the Main Controller Unit while Figure 5.2(b) shows the User Interface Unit.

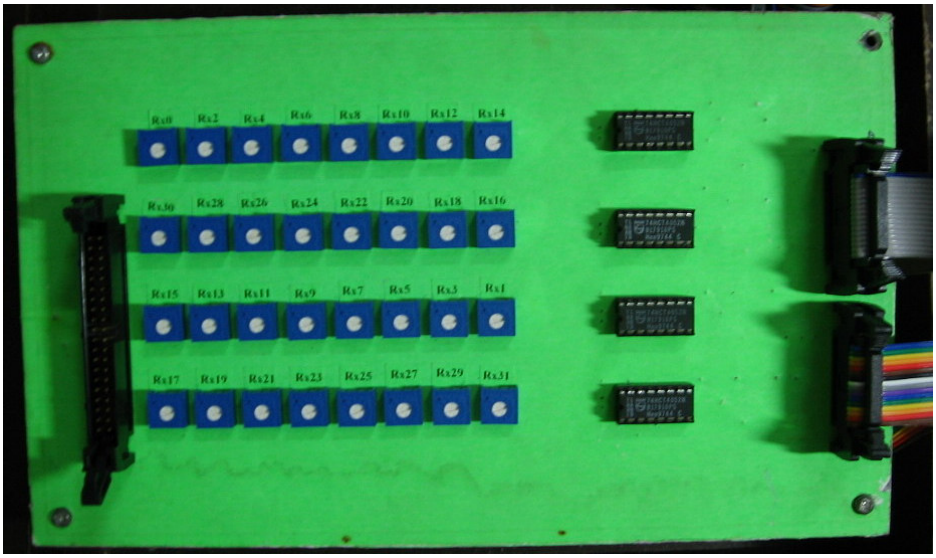


Figure 5.1(a) : Analogue Signal Conditioning PCB

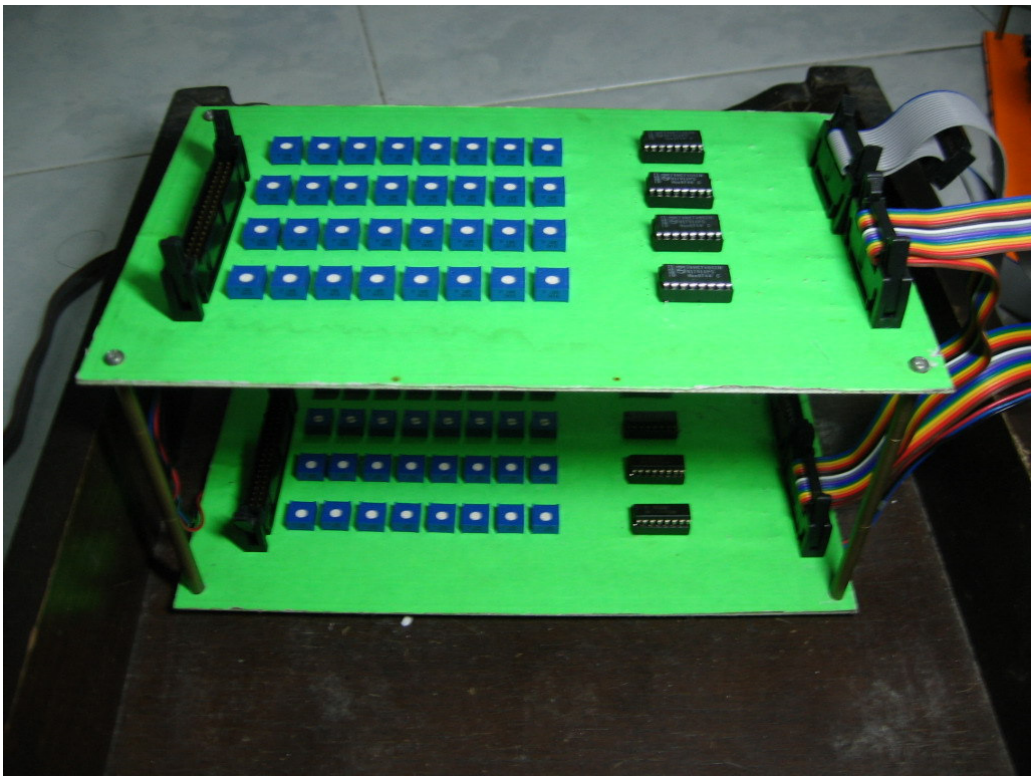


Figure 5.1(b) : Analogue Signal Conditioning PCB in 2 Layers

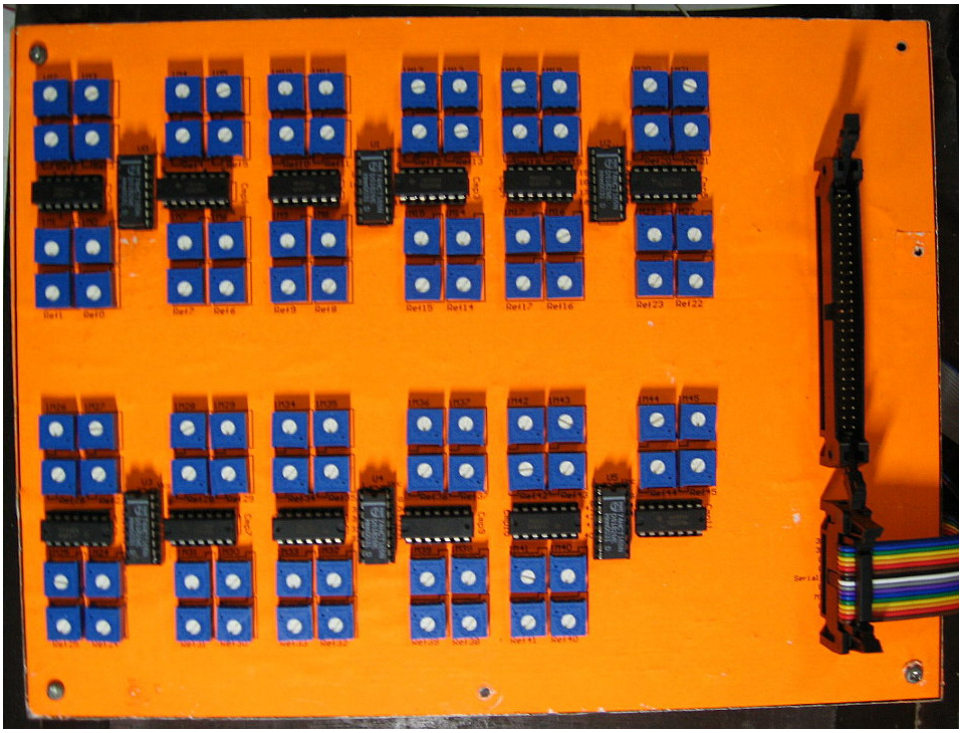


Figure 5.1(c) : Digital Signal Conditioning PCB

Figure 5.1(d) : Digital Signal Conditioning PCB in 2 layers

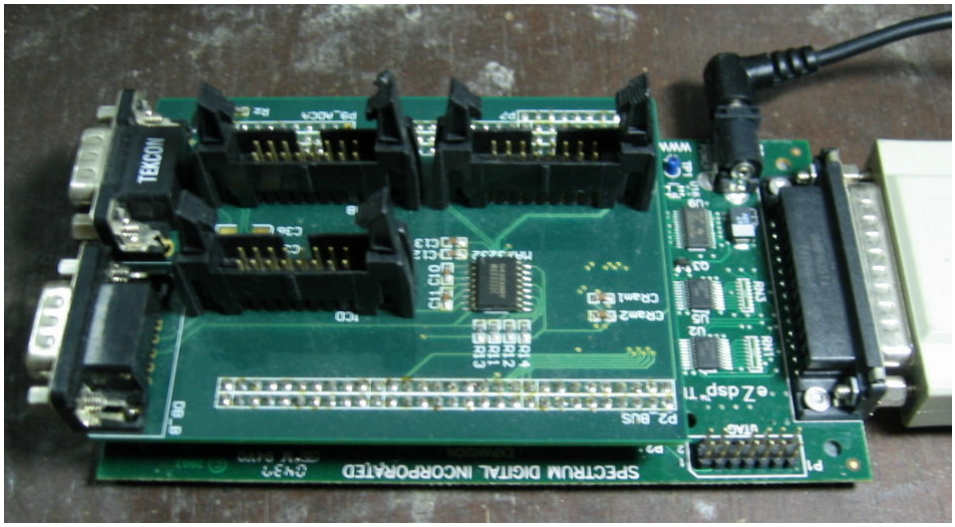


Figure 5.2(a) : Main Controller Unit

Figure 5.2(b) : User Interface Unit

5.2 Image Reconstruction

This part discusses the results obtained from the image reconstruction algorithm. The image reconstruction algorithm being used was the New Hybrid Reconstruction algorithm as explained in Section 3.3. The experiment for static object was carried out by holding statically the test object within the sensor peripheral by using a retort stand while the object was dropped from a drop distance in the dropping object experiment. Its cross sectional image is then captured using the developed system. The goals of these experiments were to determine the accuracy of the reconstructed cross sectional images compared to the cross section of the actual objects. This test assured that the data acquisition hardware system was working correctly and determined the accuracy of the data acquisition and image reconstruction algorithm. It is very important to ensure that this part is correct because the accuracy of the mass flow rate measurement is dependant to the quality of the reconstructed cross sectional images.

In order to carry out these experiments, three cylinders with difference in cross sectional area were built using card boards for use as test objects. Cylindrical shape was chosen as it most resembles the actual flow of solid materials in a pipeline (Pang, 2004a). The details of these objects are specified as in the following table. The size of the test objects was determined based on the percentage of total measurable area of the sensor fixture which is 5674.5 mm^2. Results from these experiments were used to compare with the actual results from calculations as in Table 5.1 below.

Table 5.1 : Static objects dimensions

Pipe radius: 42.5 mm, Pipe area: 5674.50 mm^2

Object	Percentage of Total Measurable Area	Cylinder Area (mm^2)	Cylinder radius (mm)
A	25%	1418.63	21.25
B	50%	2837.25	30.05
C	75%	4255.88	36.81

5.2.1 Image Reconstruction for Static Object

From the study of Pang, (2004a), object being placed at different location may result in different expected measurement due to the sensor configuration and threshold value. Therefore, an estimation of the minimum and maximum expected measurement was carried out before proceeding with online experiments. The following were the results from this experiment.

5.2.1.1 Effect of different location to Object A measurement

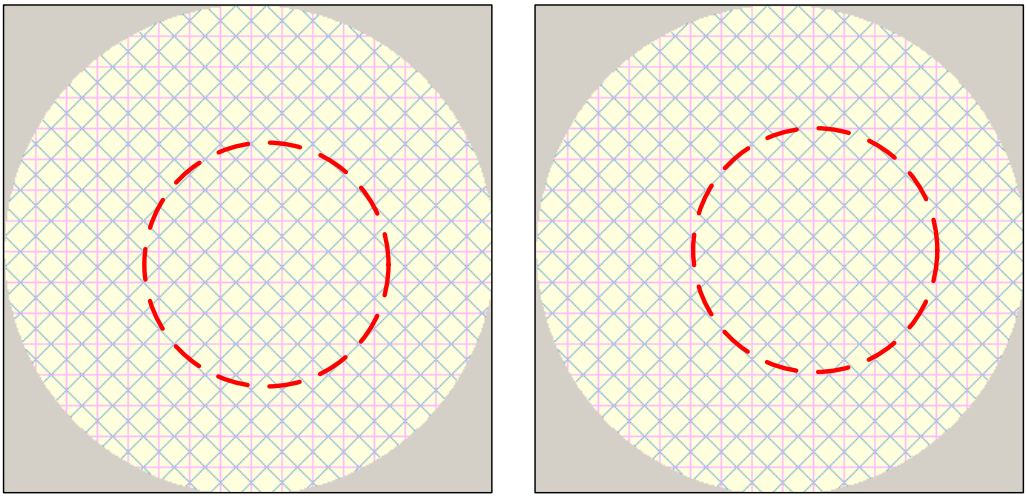


Figure 5.3 : Object A minimum and maximum measurement

From calculations, Table 5.2 was obtained and shown below:

Table 5.2 : Object A minimum and maximum measurement

	Object A Min	**Object A Max**
Expected Area	1418.89 mm^2	1428.98 mm^2
Actual Area	1418.63 mm^2	1418.63 mm^2
$Error_{(Expected - Actual)}$	-0.26 %	0.73 %

5.2.1.2 Effect of different location to Object B measurement

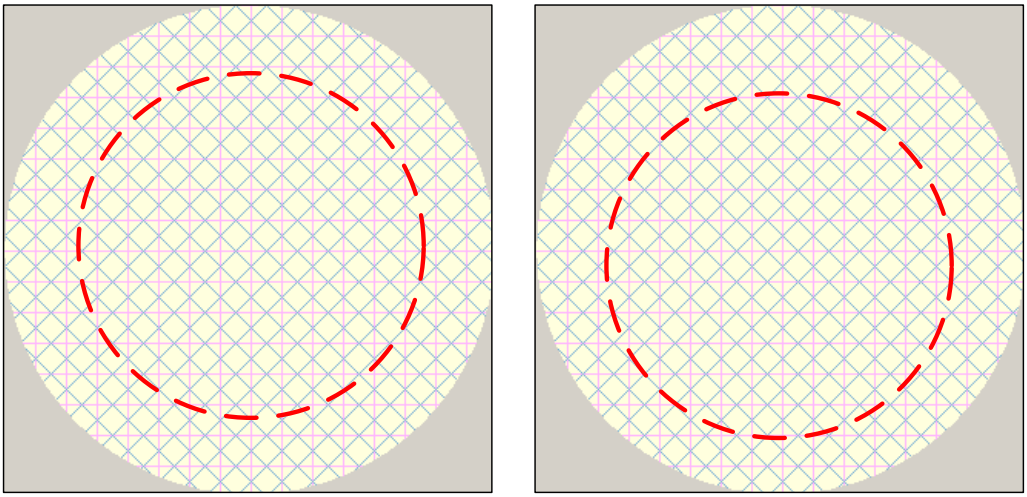


Figure 5.4 : Object B minimum and maximum measurement

From calculations, Table 5.3 was obtained and shown below:

Table 5.3 : Object B minimum and maximum measurement

	Object B Min	**Object B Max**
Expected Area	2822.73 mm^2	2865.18 mm^2
Actual Area	2837.25 mm^2	2837.25 mm^2
$Error_{(Expected - Actual)}$	-0.51 %	0.98 %

5.2.1.3 Effect of different location to Object C measurement

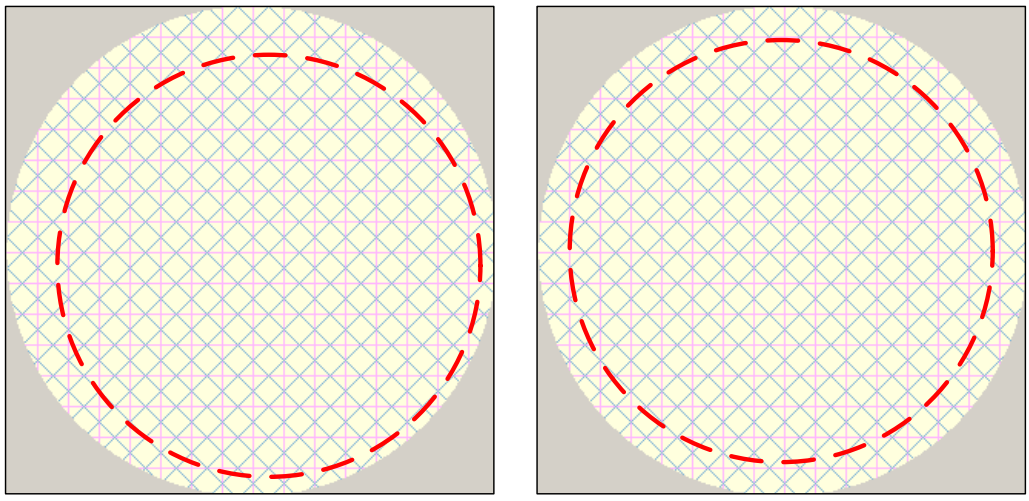


Figure 5.5 : Object C minimum and maximum measurement

From calculations, Table 5.4 was obtained and shown below:

Table 5.4 : Object C minimum and maximum measurement

	Object C Min	**Object C Max**
Expected Area	4045.39 mm^2	4315.47 mm^2
Actual Area	4255.88 mm^2	4255.88 mm^2
$Error_{(Expected - Actual)}$	-4.95 %	1.4 %

5.2.1.4 Static Object Measurement Results

In this experiment, each cylinder object was held statically inside the pipe to simulate continuous solid flow. One set of data was captured for each object in the experiments. Then, image reconstruction algorithm was applied to the data, which resulted in a reconstructed cross sectional image. The images for the static objects are presented as follows. Shown beside each images are numerical representation of the image in hexadecimal. This is followed by a table of measured concentration profile and a table of expected concentration profile.

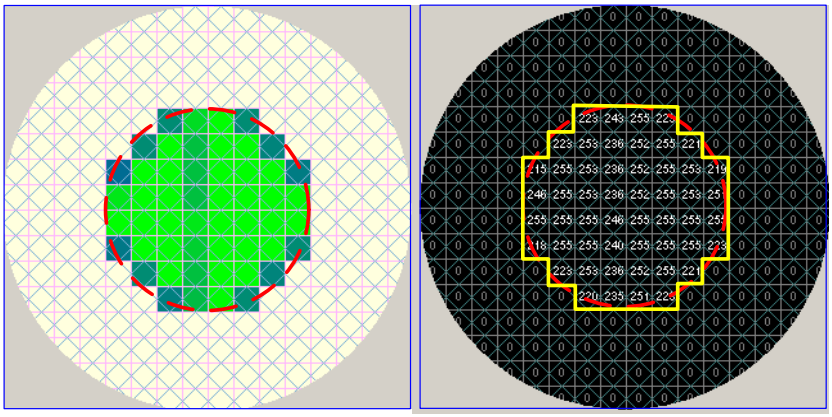


Figure 5.6 : Reconstructed image for Object A

Table 5.5 : Object A measured concentration profile

0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	24.42	26.08	27.86	24.75	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	24.75	28.08	26.19	27.97	28.30	24.53	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	24.19	28.30	28.30	26.64	28.30	28.30	28.30	24.75	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	28.30	28.30	28.30	27.30	28.30	28.30	28.30	28.30	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	27.30	28.30	28.08	26.19	27.97	28.30	28.08	27.86	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	23.86	28.30	28.08	26.19	27.97	28.30	28.08	24.30	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	24.75	28.08	26.19	27.97	28.30	24.53	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	24.75	26.97	28.30	24.75	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table 5.6 : Object A expected concentration profile

0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	24.75	28.30	28.30	24.75	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	24.75	28.30	28.30	28.30	28.30	24.75	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	24.75	28.30	28.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	24.75	28.30	28.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	24.75	28.30	28.30	28.30	28.30	24.75	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	24.75	28.30	28.30	24.75	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Actual Area Concentration : 1418.63
Expected Area Concentration : 1428.98
Measured Area Concentration : 1407.12
Error (Expected-Actual) : 0.73 %
Error (Measured-Expected) : -1.53 %

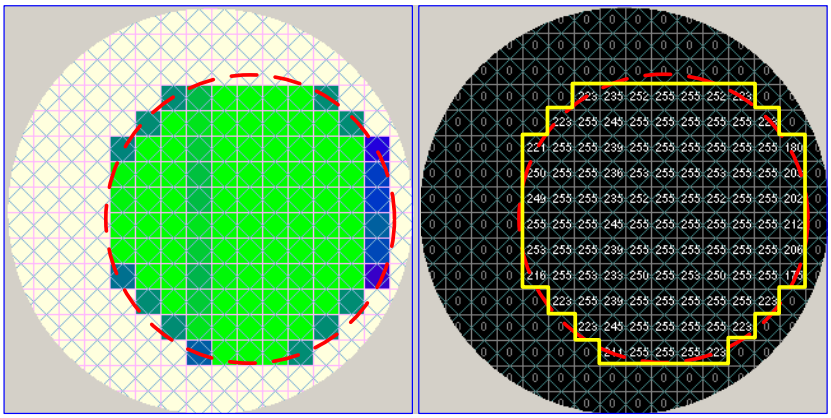


Figure 5.7 : Reconstructed image for Object B

Table 5.7 : Object B measured concentration profile

0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	23.42	28.30	28.30	28.30	24.75	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	24.75	27.19	28.30	28.30	28.30	28.30	24.75	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	24.75	28.30	26.52	28.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00
0.00	0.00	0.00	0.00	23.97	28.30	28.08	25.86	27.75	28.30	28.08	27.75	28.30	28.30	19.42	0.00
0.00	0.00	0.00	0.00	28.08	28.30	28.30	26.52	28.30	28.30	28.30	28.30	28.30	28.30	22.86	0.00
0.00	0.00	0.00	0.00	28.30	28.30	28.30	27.19	28.30	28.30	28.30	28.30	28.30	28.30	23.53	0.00
0.00	0.00	0.00	0.00	27.63	28.30	28.30	26.08	27.97	28.30	28.30	27.97	28.30	28.30	22.42	0.00
0.00	0.00	0.00	0.00	27.75	28.30	28.30	26.19	28.08	28.30	28.30	28.08	28.30	28.30	22.53	0.00
0.00	0.00	0.00	0.00	24.53	28.30	28.30	26.52	28.30	28.30	28.30	28.30	28.30	28.30	19.98	0.00
0.00	0.00	0.00	0.00	0.00	24.75	28.30	27.19	28.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	24.75	26.08	27.97	28.30	28.30	27.97	24.75	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table 5.8 : Object B expected concentration profile

0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	24.75	28.30	28.30	28.30	24.75	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	24.75	28.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	24.75	28.30	28.30	28.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00
0.00	0.00	0.00	0.00	24.75	28.30	28.08	28.30	28.30	28.30	28.30	28.30	28.30	28.30	24.75	0.00
0.00	0.00	0.00	0.00	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	0.00
0.00	0.00	0.00	0.00	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	0.00
0.00	0.00	0.00	0.00	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	0.00
0.00	0.00	0.00	0.00	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	0.00
0.00	0.00	0.00	0.00	24.75	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	24.75	0.00
0.00	0.00	0.00	0.00	0.00	24.75	28.30	28.30	28.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	24.75	28.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Actual Area Concentration : 2837.25
Expected Area Concentration : 2864.96
Measured Area Concentration : 2808.47
Error (Expected-Actual) : 0.98 %
Error (Measured-Expected) : -1.97 %

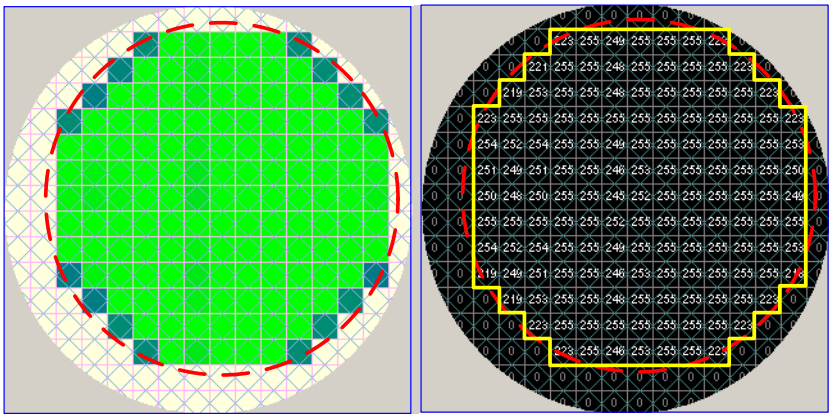


Figure 5.8 : Reconstructed image for Object C

Table 5.9 : Object C measured concentration profile

0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	24.75	28.30	27.30	28.08	28.30	28.30	24.75	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	24.75	28.30	28.30	28.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00	0.00
0.00	0.00	0.00	24.30	28.08	28.30	28.30	27.52	28.30	28.30	28.30	28.30	24.75	0.00	0.00	0.00
0.00	0.00	24.30	27.63	27.86	28.30	28.30	27.30	28.08	28.30	28.30	28.30	28.30	28.30	24.19	0.00
0.00	0.00	28.19	27.97	28.19	28.30	28.30	27.63	28.30	28.30	28.30	28.30	28.30	28.30	28.08	0.00
0.00	0.00	28.30	28.30	28.30	28.30	28.30	27.97	28.30	28.30	28.30	28.30	28.30	28.30	28.30	0.00
0.00	0.00	27.75	27.52	27.75	28.30	28.30	27.19	27.97	28.30	28.30	28.30	28.30	28.30	27.63	0.00
0.00	0.00	27.86	27.63	27.86	28.30	28.30	27.30	28.08	28.30	28.30	28.30	28.30	28.30	27.75	0.00
0.00	0.00	28.19	27.97	28.19	28.30	28.30	27.63	28.30	28.30	28.30	28.30	28.30	28.30	28.08	0.00
0.00	0.00	24.75	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	24.66	0.00
0.00	0.00	0.00	24.30	28.08	28.30	28.30	27.52	28.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00
0.00	0.00	0.00	0.00	24.53	28.30	28.30	27.52	28.30	28.30	28.30	28.30	24.75	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	24.75	28.30	27.63	28.30	28.30	28.30	24.66	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table 5.10 : Object C expected concentration profile

0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	24.75	28.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	24.75	28.30	28.30	28.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00	0.00
0.00	0.00	0.00	24.75	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00
0.00	0.00	24.75	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	24.75	0.00
0.00	0.00	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	0.00
0.00	0.00	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	0.00
0.00	0.00	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	0.00
0.00	0.00	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	0.00
0.00	0.00	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	0.00
0.00	0.00	24.75	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	24.66	0.00
0.00	0.00	0.00	24.75	28.08	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00
0.00	0.00	0.00	0.00	24.53	28.30	28.30	28.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	24.75	28.30	27.63	28.30	28.30	28.30	24.66	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Actual Area Concentration : 4255.88
Expected Area Concentration : 4045.39
Measured Area Concentration : 4026.86
Error (Expected-Actual) : -4.95 %
Error (Measured-Expected) : -0.46 %

The following table summarizes the results obtained.

Table 5.11 : Summary of image reconstruction results for static objects.

Object	**Percentage of Total Measurable Area** (%)	**Actual Object Area** (mm^2)	**Expected Object Area** (mm^2)	**Measured Object Area** (mm^2)	**Error** (Expected-Actual)	**Error** (Measured-Expected)
A	25	1418.63	1428.98	1407.12	0.73%	-1.53%
B	50	2837.25	2864.96	2808.47	0.98%	-1.97%
C	75	4255.88	4045.39	4026.86	-4.95%	-0.46%

In table 5.11, the measurement errors were calculated according to equation 5.1 and Equation 5.2.

$$\text{Error}_{(\text{Expected-Actual})}\ (\%) = \frac{\text{Expected Area - Actual Area}}{\text{Actual Area}}\ \text{x}\ 100\ \% \qquad \text{...5.1}$$

$$\text{Error}_{(\text{Measured-Expected})}\ (\%) = \frac{\text{Measured Area - Expected Area}}{\text{Expected Area}}\ \text{x}\ 100\ \% \qquad \text{...5.2}$$

5.2.1.5 Discussion

From the study of effects of different locations of object to the measurement results, it was noted that the expected area measurements were more accurate to actual area when the object is smaller and placed at the pixels that fully block the light beams. On the other hand, when the object was placed at location in between two light beams but not equally spaced, the expected measurement was lower than the actual area. In this situation, there were little areas that does not block the light beam sufficiently (sensor reading below threshold) and causes these areas undetected. Therefore, as the object became bigger, the total of 'little areas' also became more. As a result, the expected minimum measurement for larger objects tends to be lower as the object becomes larger.

During the experiments, the test object was placed into the sensor fixture and measurements were taken. The results obtained support the above discussion whereby the results were either close to expected minimum measurement or expected maximum measurement. The error between the actual measurement and expected measurement were also very small (less than 2%). This proved that the data acquisition and image reconstruction algorithm is capable of acquiring accurate data to reconstruct cross sectional images with only little error compared to the expected measurements.

5.2.2 Image Reconstruction for Dropping Object

These experiments aimed to verify that the images reconstructed during object drop were correct and the same as taken during static object measurement. In this part of experiment, the same three objects were used to represent solid flow which were Object A (25%), Object B (50%) and Object C (75%). For each object, the experiment was carried out by dropping it from three different drop distances. The experiment setting is shown in Figure 5.9. The same setting was used for all other experiments that followed.

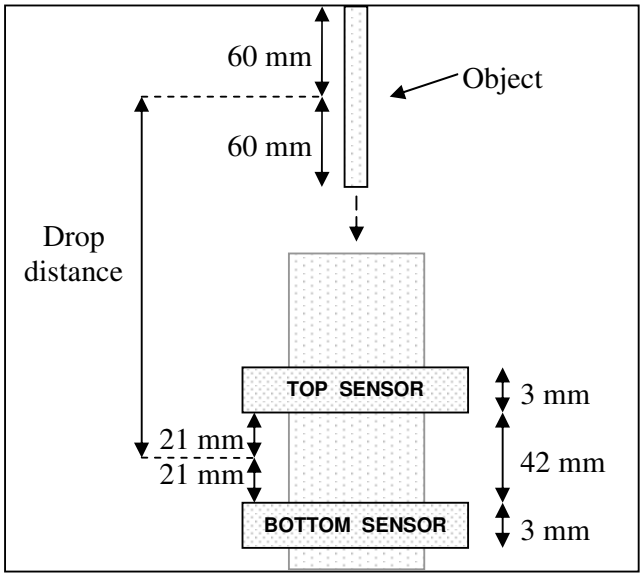


Figure 5.9 : Setting for all experiments

For all experiments, the actual values of the cross sectional areas of the three objects used are as follows:

Table 5.12 : Actual cross sectional areas

	Object A	**Object B**	**Object C**
Actual Cross Sectional Area	1418.63 mm^2	2837.25 mm^2	4255.88 mm^2
Percentage of Total Area	25 %	50 %	75 %

In the following, results obtained from the experiments are presented. The image reconstructed for each object drop is shown first, followed by the measured concentration profile and the expected concentration profile. Red circle represent the actual object location while the yellow shape represent the expected image or shape from image reconstruction algorithm. The values in the numerical tomogram are sensor values (before conversion to area) while the values in the tabulated data are values in unit of mm^2.

5.2.2.1 Image Reconstruction for 225mm drop distance

In this set of experiments, the object cross sectional is captured when Object A, Object B and Object C were dropped one by one from a drop distance of 225 mm.

Object A - 225 mm drop distance

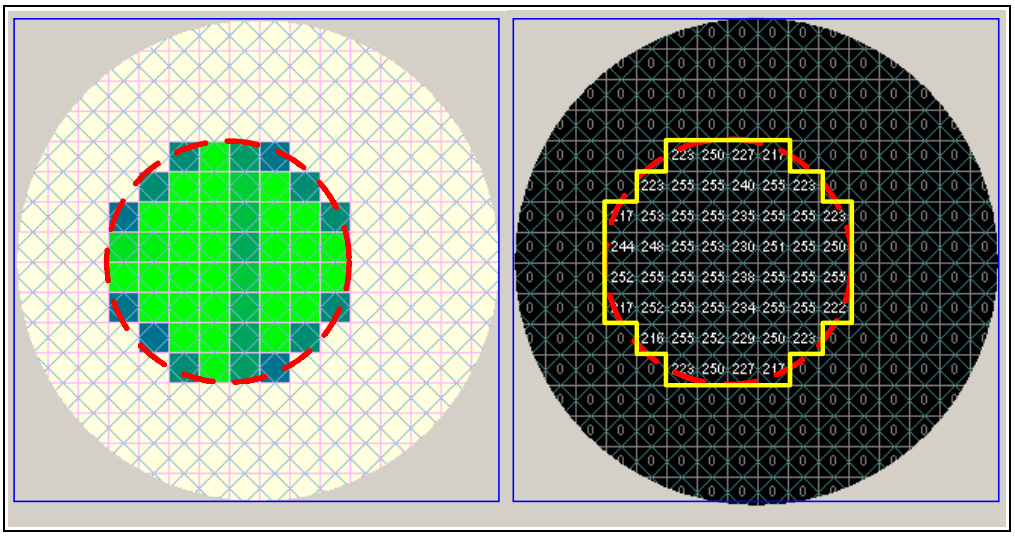


Figure 5.10 : Object A reconstructed image from 225 mm drop distance

Table 5.13 : Object A measured concentration profile from 225 mm drop distance

0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	24.75	27.75	25.19	24.08	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	23.97	28.30	27.97	25.41	27.75	24.75	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	24.08	27.97	28.30	28.30	25.97	28.30	28.30	24.64	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	27.97	28.30	28.30	28.30	26.41	28.30	28.30	28.30	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	27.08	27.52	28.30	28.08	25.53	27.86	28.30	27.75	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	24.08	28.08	28.30	28.30	26.08	28.30	28.30	24.75	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	24.75	28.30	28.30	26.64	28.30	24.75	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	24.75	27.75	25.19	24.08	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Total concentration : *1399.35 mm*^{2}

Table 5.14 : Object A expected concentration profile from 225 mm drop distance

0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	24.75	28.30	28.30	24.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	24.75	28.30	28.30	28.30	28.30	24.75	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	24.75	28.30	28.30	28.30	28.30	28.30	28.30	24.64	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	24.75	28.30	28.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	24.75	28.30	28.30	28.30	28.30	24.75	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	24.75	28.30	28.30	24.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Total concentration : *1428.87mm*^{2}

% Error $_{\text{(Expected-Actual)}}$ = 0.72 %

% Error $_{\text{(Measured-Expected)}}$ = -2.07 %

Object B - 225 mm drop distance

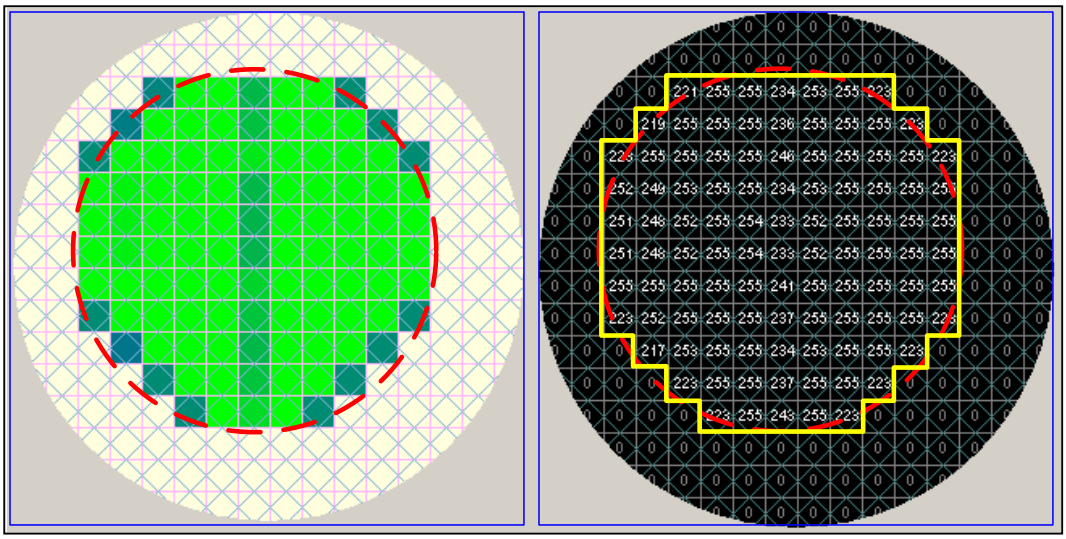


Figure 5.11 : Object B reconstructed image from 225 mm drop distance

Table 5.15 : Object B measured concentration profile from 225 mm drop distance

0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	24.75	28.30	26.97	28.30	24.75	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	24.75	28.30	28.30	26.30	28.30	28.30	24.75	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	24.08	28.08	28.30	28.30	25.97	28.08	28.30	28.30	24.75	0.00	0.00	0.00	0.00
0.00	0.00	24.75	27.97	28.30	28.30	28.30	26.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00	0.00
0.00	0.00	28.30	28.30	28.30	28.30	28.30	26.75	28.30	28.30	28.30	28.30	28.30	0.00	0.00	0.00
0.00	0.00	27.86	27.52	27.97	28.30	28.19	25.86	27.97	28.30	28.30	28.30	28.30	0.00	0.00	0.00
0.00	0.00	27.86	27.52	27.97	28.30	28.19	25.86	27.97	28.30	28.30	28.30	28.30	0.00	0.00	0.00
0.00	0.00	27.97	27.63	28.08	28.30	28.30	25.97	28.08	28.30	28.30	28.30	28.30	0.00	0.00	0.00
0.00	0.00	24.75	28.30	28.30	28.30	28.30	27.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00	0.00
0.00	0.00	0.00	24.30	28.30	28.30	28.30	26.19	28.30	28.30	28.30	24.75	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	24.53	28.30	28.30	25.97	28.08	28.30	24.75	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Total concentration : *2835.55 mm*^{2}

Table 5.16 : Object B expected concentration profile from 225 mm drop distance

0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	24.75	28.30	28.30	28.30	24.75	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	24.75	28.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	24.75	28.08	28.30	28.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00	0.00	0.00
0.00	0.00	24.75	28.30	28.30	28.30	28.30	26.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00	0.00
0.00	0.00	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	0.00	0.00	0.00
0.00	0.00	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	0.00	0.00	0.00
0.00	0.00	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	0.00	0.00	0.00
0.00	0.00	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	0.00	0.00	0.00
0.00	0.00	24.75	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00	0.00
0.00	0.00	0.00	24.75	28.30	28.30	28.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	24.75	28.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Total concentration : 2862.96 mm^2

% Error $_{(Expected-Actual)}$ = 0.91 %

% Error $_{(Measured-Expected)}$ = -0.96 %

Object C - 225 mm drop distance

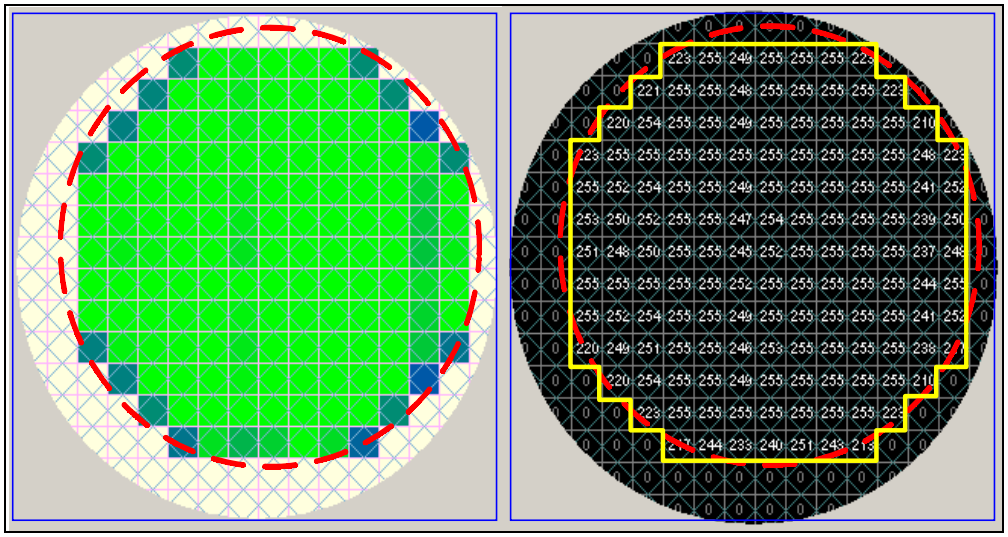


Figure 5.12 : Object C reconstructed image from 225 mm drop distance

Table 5.17 : Object C measured concentration profile from 225 mm drop distance

0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	24.08	27.08	25.86	26.64	27.86	26.97	23.64	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	24.75	28.30	28.30	28.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00	0.00
0.00	0.00	0.00	24.42	28.19	28.30	28.30	27.63	28.30	28.30	28.30	28.30	28.30	23.31	0.00	0.00
0.00	0.00	24.42	27.63	27.86	28.30	28.30	27.30	28.08	28.30	28.30	28.30	28.30	26.41	24.08	0.00
0.00	0.00	28.30	27.97	28.19	28.30	28.30	27.63	28.30	28.30	28.30	28.30	28.30	26.75	27.97	0.00
0.00	0.00	28.30	28.30	28.30	28.30	28.30	27.97	28.30	28.30	28.30	28.30	28.30	27.08	28.30	0.00
0.00	0.00	27.86	27.52	27.75	28.30	28.30	27.19	27.97	28.30	28.30	28.30	28.30	26.30	27.52	0.00
0.00	0.00	28.08	27.75	27.97	28.30	28.30	27.41	28.19	28.30	28.30	28.30	28.30	26.52	27.75	0.00
0.00	0.00	28.30	27.97	28.19	28.30	28.30	27.63	28.30	28.30	28.30	28.30	28.30	26.75	27.97	0.00
0.00	0.00	24.75	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	27.52	24.66	0.00
0.00	0.00	0.00	24.42	28.19	28.30	28.30	27.63	28.30	28.30	28.30	28.30	28.30	23.31	0.00	0.00
0.00	0.00	0.00	0.00	24.53	28.30	28.30	27.52	28.30	28.30	28.30	28.30	24.75	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	24.75	28.30	27.63	28.30	28.30	28.30	24.66	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Total concentration : 4006.88mm^2

Table 5.18 : Object C expected concentration profile from 225 mm drop distance

0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	24.75	28.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	24.75	28.30	28.30	28.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00	0.00
0.00	0.00	0.00	24.75	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00
0.00	0.00	24.75	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	24.75	0.00
0.00	0.00	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	0.00
0.00	0.00	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	0.00
0.00	0.00	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	0.00
0.00	0.00	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	0.00
0.00	0.00	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	0.00
0.00	0.00	24.75	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	24.66	0.00
0.00	0.00	0.00	24.75	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00
0.00	0.00	0.00	0.00	24.75	28.30	28.30	28.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	24.75	28.30	28.30	28.30	28.30	28.30	24.66	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Total concentration : 4046.50 mm^2

% Error $_{(Expected-Actual)}$ = -4.92 %

% Error $_{(Measured-Expected)}$ = -0.98%

5.2.2.2 Image Reconstruction for 275mm drop distance

In this set of experiments, the object cross sectional is captured when Object A, Object B and Object C were dropped one by one from a drop distance of 275 mm.

Object A - 275 mm drop distance

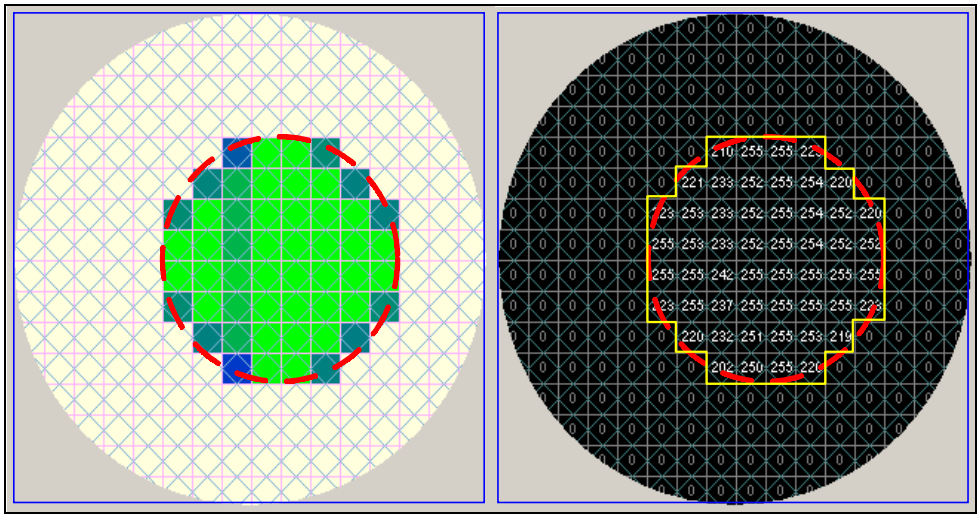


Figure 5.13 : Object A reconstructed image from 275 mm drop distance

Table 5.19 : Object A measured concentration profile from 275 mm drop distance

0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	22.42	27.75	28.30	24.42	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	24.42	25.75	27.86	28.30	28.08	24.30	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	24.75	28.30	26.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	28.30	28.30	26.86	28.30	28.30	28.30	28.30	28.30	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	28.30	28.08	25.86	27.97	28.30	28.19	27.97	27.97	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	24.75	28.08	25.86	27.97	28.30	28.19	27.97	24.42	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	24.53	25.86	27.97	28.30	28.19	24.42	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	23.31	28.30	28.30	24.75	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Total concentration : 1405.90 mm^2

Table 5.20 : Object A expected concentration profile from 275 mm drop distance

0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	24.75	28.30	28.30	24.75	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	24.75	28.30	28.30	28.30	28.30	24.75	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	24.75	28.30	28.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	24.75	28.30	28.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	24.75	28.30	28.30	28.30	28.30	24.75	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	24.75	28.30	28.30	24.75	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Total concentration : 1429.98 mm^2

% Error $_{(\text{Expected-Actual})}$ = 0.73 %

% Error $_{(\text{Measured-Expected})}$ = -1.62 %

Object B - 275 mm drop distance

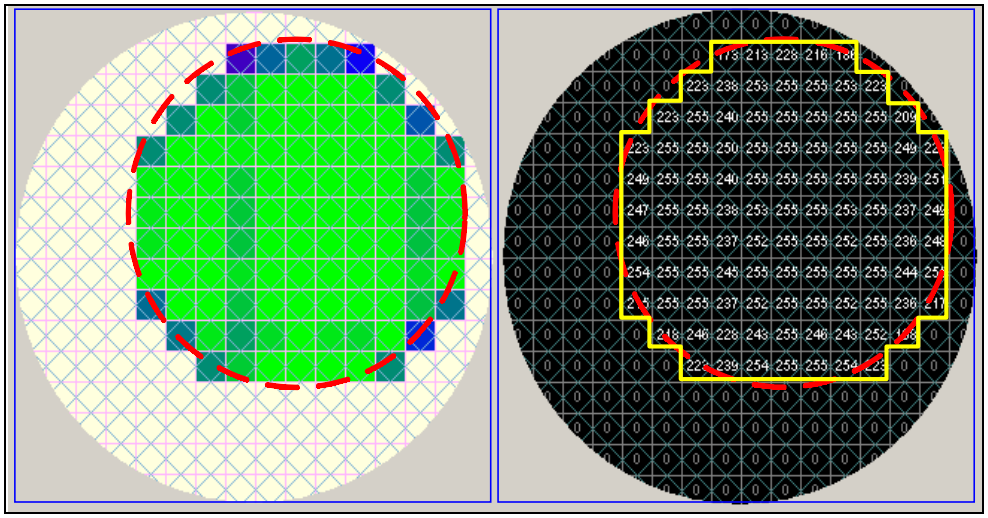


Figure 5.14 : Object B reconstructed image from 275 mm drop distance

Table 5.21 : Object B measured concentration profile from 275 mm drop distance

0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	24.75	26.52	28.19	28.30	28.30	28.19	24.75	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	24.19	27.30	25.30	26.97	28.30	27.30	26.97	27.97	21.97	0.00	0.00
0.00	0.00	0.00	0.00	23.86	28.30	28.30	26.30	27.97	28.30	28.30	27.97	28.30	26.19	24.08	0.00
0.00	0.00	0.00	0.00	28.19	28.30	28.30	27.19	28.30	28.30	28.30	28.30	28.30	27.08	28.30	0.00
0.00	0.00	0.00	0.00	27.30	28.30	28.30	26.30	27.97	28.30	28.30	27.97	28.30	26.19	27.52	0.00
0.00	0.00	0.00	0.00	27.41	28.30	28.30	26.41	28.08	28.30	28.30	28.08	28.30	26.30	27.63	0.00
0.00	0.00	0.00	0.00	27.63	28.30	28.30	26.64	28.30	28.30	28.30	28.30	28.30	26.52	27.86	0.00
0.00	0.00	0.00	0.00	24.75	28.30	28.30	27.75	28.30	28.30	28.30	28.30	28.30	27.63	24.66	0.00
0.00	0.00	0.00	0.00	0.00	24.75	28.30	26.64	28.30	28.30	28.30	28.30	28.30	23.19	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	24.75	26.41	28.08	28.30	28.30	28.08	24.75	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	19.20	23.64	25.30	23.97	20.57	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Total concentration : 2808.56 mm^2

Table 5.22 : Object B expected concentration profile from 275 mm drop distance

0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	24.75	28.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	24.75	28.30	28.30	28.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00
0.00	0.00	0.00	0.00	24.75	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	24.75	0.00
0.00	0.00	0.00	0.00	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	0.00
0.00	0.00	0.00	0.00	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	0.00
0.00	0.00	0.00	0.00	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	0.00
0.00	0.00	0.00	0.00	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	0.00
.00	0.00	0.00	0.00	24.75	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	24.66	0.00
0.00	0.00	0.00	0.00	0.00	24.75	28.30	28.30	28.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	24.75	28.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	24.75	28.30	28.30	28.30	24.66	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Total concentration : 2865.01 mm^2

% Error $_{(Expected-Actual)}$ = 0.98 %

% Error $_{(Measured-Expected)}$ = -2.35 %

Object C - 275 mm drop distance

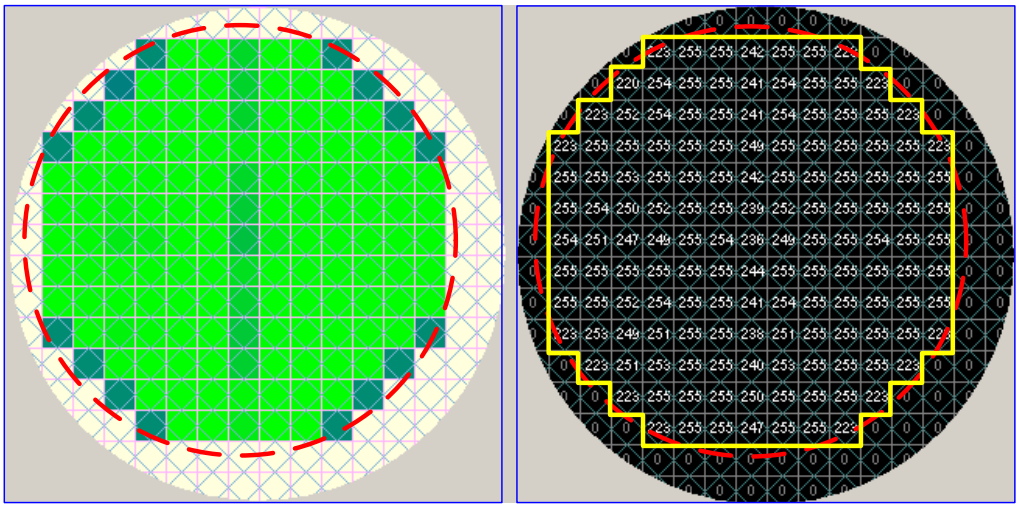


Figure 5.15 : Object C Reconstructed Image from 275 mm drop distance

Table 5.23 : Object C measured concentration profile from 275 mm drop distance

0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	24.75	28.30	28.30	27.41	28.30	28.30	24.75	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	24.75	28.30	28.30	28.30	27.75	28.30	28.30	28.30	24.75	0.00	0.00	0.00	0.00
0.00	0.00	24.75	27.86	28.08	28.30	28.30	26.64	28.08	28.30	28.30	28.30	24.75	0.00	0.00	0.00
0.00	24.75	28.08	27.63	27.86	28.30	28.30	26.41	27.86	28.30	28.30	28.30	28.30	24.75	0.00	0.00
0.00	28.30	28.30	27.97	28.19	28.30	28.30	26.75	28.19	28.30	28.30	28.30	28.30	28.30	0.00	0.00
0.00	28.30	28.30	28.30	28.30	28.30	28.30	27.08	28.30	28.30	28.30	28.30	28.30	28.30	0.00	0.00
0.00	28.19	27.86	27.41	27.63	28.30	28.19	26.19	27.63	28.30	28.30	28.19	28.30	28.30	0.00	0.00
0.00	28.30	28.19	27.75	27.97	28.30	28.30	26.52	27.97	28.30	28.30	28.30	28.30	28.30	0.00	0.00
0.00	28.30	28.30	28.08	28.30	28.30	28.30	26.86	28.30	28.30	28.30	28.30	28.30	28.30	0.00	0.00
0.00	24.66	28.30	28.30	28.30	28.30	28.30	27.63	28.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00
0.00	0.00	24.75	27.97	28.19	28.30	28.30	26.75	28.19	28.30	28.30	28.30	24.75	0.00	0.00	0.00
0.00	0.00	0.00	24.42	28.19	28.30	28.30	26.75	28.19	28.30	28.30	24.75	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	24.66	28.30	28.30	26.86	28.30	28.30	24.75	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Total concentration : 4019.31 mm^2

Table 5.24 : Object C expected concentration profile from 275 mm drop distance

0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	24.75	28.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	24.75	28.30	28.30	28.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00	0.00	0.00
0.00	0.00	24.75	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00	0.00
0.00	24.75	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00
0.00	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	0.00	0.00
0.00	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	0.00	0.00
0.00	28.19	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	0.00	0.00
0.00	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	0.00	0.00
0.00	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	0.00	0.00
0.00	24.66	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00
0.00	0.00	24.75	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00	0.00
0.00	0.00	0.00	24.75	28.30	28.30	28.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	24.66	28.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Total concentration : 4046.39 mm^2

% Error $_{(Expected-Actual)}$ = -4.92 %

% Error $_{(Measured-Expected)}$ = -0.67 %

5.2.2.3 Image Reconstruction for 350mm drop distance

In this set of experiments, the object cross sectional is captured when Object A, Object B and Object C were dropped one by one from a drop distance of 350 mm.

Object A - 350 mm drop distance

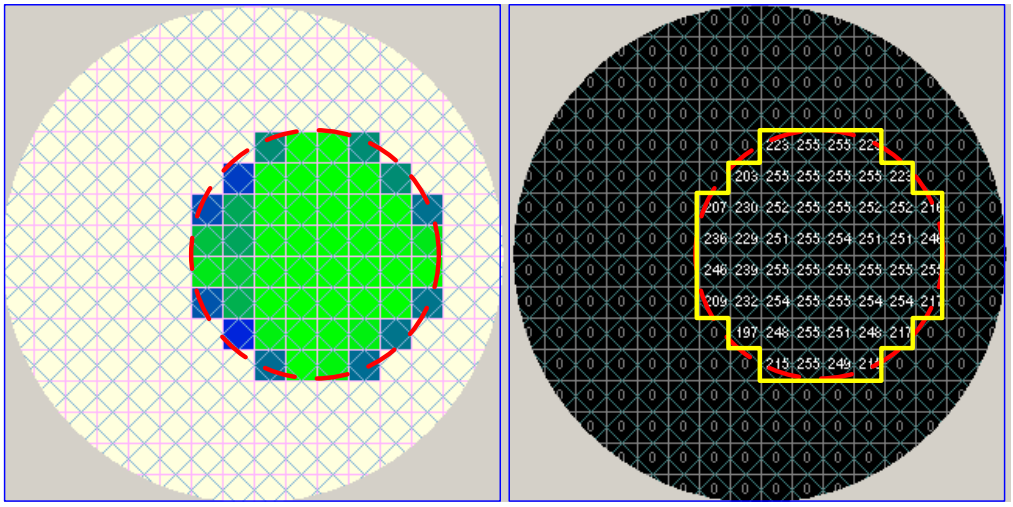


Figure 5.16 : Object A Reconstructed Image from 350 mm drop distance

Table 5.25 : Object A measured concentration profile from 350 mm drop distance

0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	23.86	28.30	27.63	23.86	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	21.86	27.52	28.30	27.86	27.52	24.08	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	23.19	25.75	28.19	28.30	28.30	28.19	28.19	24.08	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	27.30	26.52	28.30	28.30	28.30	28.30	28.30	28.30	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	26.19	25.41	27.86	28.30	28.19	27.86	27.86	27.30	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	22.97	25.53	27.97	28.30	28.30	27.97	27.97	23.97	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	22.53	28.30	28.30	28.30	28.30	24.75	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	24.75	28.30	28.30	24.75	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Total concentration : 1397.13 mm^2

Table 5.26 : Object A expected concentration profile from 350 mm drop distance

0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	24.75	28.30	28.30	24.75	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	24.75	28.30	28.30	28.30	28.30	24.75	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	24.75	28.30	28.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	24.75	28.30	28.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	24.75	28.30	28.30	28.30	28.30	24.75	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	24.75	28.30	28.30	24.75	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Total concentration : 1428.98 mm^2

% Error $_{(\text{Expected-Actual})}$ = 0.73 %

% Error $_{(\text{Measured-Expected})}$ = -2.23 %

Object B - 350 mm drop distance

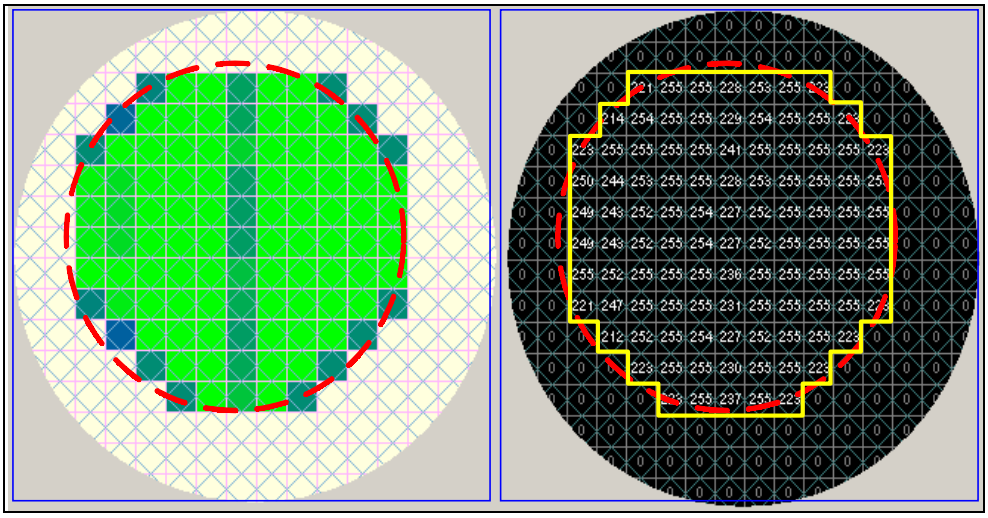


Figure 5.17 : Object B Reconstructed Image from 350 mm drop distance

Table 5.27 : Object B measured concentration profile from 350 mm drop distance

0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	24.75	28.30	26.30	28.30	24.75	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	24.75	28.30	28.30	25.53	28.30	28.30	24.75	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	23.53	27.97	28.30	28.19	25.19	27.97	28.30	28.30	24.75	0.00	0.00	0.00	0.00
0.00	0.00	24.53	27.41	28.30	28.30	28.30	25.64	28.30	28.30	28.30	28.30	24.75	0.00	0.00	0.00
0.00	0.00	28.30	27.97	28.30	28.30	28.30	26.19	28.30	28.30	28.30	28.30	28.30	0.00	0.00	0.00
0.00	0.00	27.63	26.97	27.97	28.30	28.19	25.19	27.97	28.30	28.30	28.30	28.30	0.00	0.00	0.00
0.00	0.00	27.63	26.97	27.97	28.30	28.19	25.19	27.97	28.30	28.30	28.30	28.30	0.00	0.00	0.00
0.00	0.00	27.75	27.08	28.08	28.30	28.30	25.30	28.08	28.30	28.30	28.30	28.30	0.00	0.00	0.00
0.00	0.00	24.75	28.30	28.30	28.30	28.30	26.75	28.30	28.30	28.30	28.30	24.75	0.00	0.00	0.00
0.00	0.00	0.00	23.75	28.19	28.30	28.30	25.41	28.19	28.30	28.30	24.75	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	24.53	28.30	28.30	25.30	28.08	28.30	24.75	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Total concentration : 2823.01 mm^2

Table 5.28 : Object B expected concentration profile from 350 mm drop distance

0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	24.75	28.30	28.30	28.30	24.75	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	24.75	28.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	24.75	28.30	28.30	28.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00	0.00	0.00
0.00	0.00	24.75	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00	0.00
0.00	0.00	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	0.00	0.00	0.00
0.00	0.00	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	0.00	0.00	0.00
0.00	0.00	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	0.00	0.00	0.00
0.00	0.00	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	0.00	0.00	0.00
0.00	0.00	24.75	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00	0.00
0.00	0.00	0.00	24.75	28.30	28.30	28.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	24.75	28.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Total concentration : 2865.18 mm^2

% Error $_{(Expected-Actual)}$ = 0.98 %

% Error $_{(Measured-Expected)}$ = -1.47 %

Object C - 350 mm drop distance

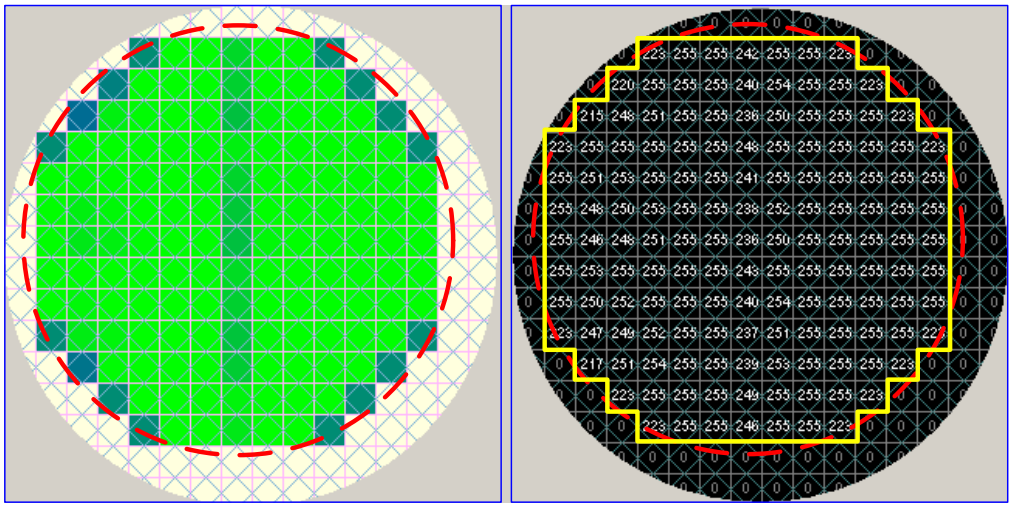


Figure 5.18 : Object C Reconstructed Image from 350 mm drop distance

Table 5.29 : Object C measured concentration profile from 350 mm drop distance

0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	24.75	28.30	28.30	27.30	28.30	28.30	24.75	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	24.75	28.30	28.30	28.30	27.63	28.30	28.30	28.30	24.75	0.00	0.00	0.00	0.00
0.00	0.00	24.08	27.86	28.19	28.30	28.30	26.52	28.08	28.30	28.30	28.30	24.75	0.00	0.00	0.00
0.00	24.75	27.41	27.63	27.97	28.30	28.30	26.30	27.86	28.30	28.30	28.30	28.30	24.75	0.00	0.00
0.00	28.30	27.75	27.97	28.30	28.30	28.30	26.64	28.19	28.30	28.30	28.30	28.30	28.30	0.00	0.00
0.00	28.30	28.08	28.30	28.30	28.30	28.30	26.97	28.30	28.30	28.30	28.30	28.30	28.30	0.00	0.00
0.00	28.30	27.30	27.52	27.86	28.30	28.30	26.19	27.75	28.30	28.30	28.30	28.30	28.30	0.00	0.00
0.00	28.30	27.52	27.75	28.08	28.30	28.30	26.41	27.97	28.30	28.30	28.30	28.30	28.30	0.00	0.00
0.00	28.30	27.86	28.08	28.30	28.30	28.30	26.75	28.30	28.30	28.30	28.30	28.30	28.30	0.00	0.00
0.00	24.66	28.30	28.30	28.30	28.30	28.30	27.52	28.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00
0.00	0.00	23.86	27.52	27.86	28.30	28.30	26.19	27.75	28.30	28.30	28.30	24.75	0.00	0.00	0.00
0.00	0.00	0.00	24.42	28.30	28.30	28.30	26.64	28.19	28.30	28.30	24.75	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	24.66	28.30	28.30	26.86	28.30	28.30	24.75	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Total concentration : 4013.10 mm^2

Table 5.30 : Object C expected concentration profile from 350 mm drop distance

0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	24.75	28.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	24.75	28.30	28.30	28.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00	0.00	0.00
0.00	0.00	24.75	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00	0.00
0.00	24.75	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00
0.00	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	0.00	0.00
0.00	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	0.00	0.00
0.00	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	0.00	0.00
0.00	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	0.00	0.00
0.00	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	0.00	0.00
0.00	24.66	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00
0.00	0.00	24.75	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00	0.00
0.00	0.00	0.00	24.75	28.30	28.30	28.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	24.66	28.30	28.30	28.30	28.30	28.30	24.75	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Total concentration : 4046.50 mm^2

% Error $_{(\text{Expected-Actual})}$ = -4.92 %

% Error $_{(\text{Measured-Expected})}$ = -0.83 %

5.2.2.4 Results summary of image reconstruction for dropping object

The image reconstruction results obtained from the dropping objects are tabulated in the following Table 5.31 while the graphical representation is shown in Figure 5.19.

Table 5.31 : Summary of image reconstruction results obtained from dropping objects

	drop distance (mm)	Measured Area (mm^2)	Expected Area (mm^2)	Actual Area (mm^2)	% Error (Expected-Actual)	% Error (Measured-Expected)
A	225	1399.35	1428.87	1418.63	0.72%	-2.07%
	275	1405.90	1428.98	1418.63	0.73%	-1.62%
	350	1397.13	1428.98	1418.63	0.73%	-2.23%
B	225	2835.55	2862.96	2837.25	0.91%	-0.96%
	275	2797.54	2865.01	2837.25	0.98%	-2.35%
	350	2823.01	2865.18	2837.25	0.98%	-1.47%
C	225	4006.88	4046.50	4255.88	-4.92%	-0.98%
	275	4019.31	4046.39	4255.88	-4.92%	-0.67%
	350	4013.10	4046.50	4255.88	-4.92%	-0.83%

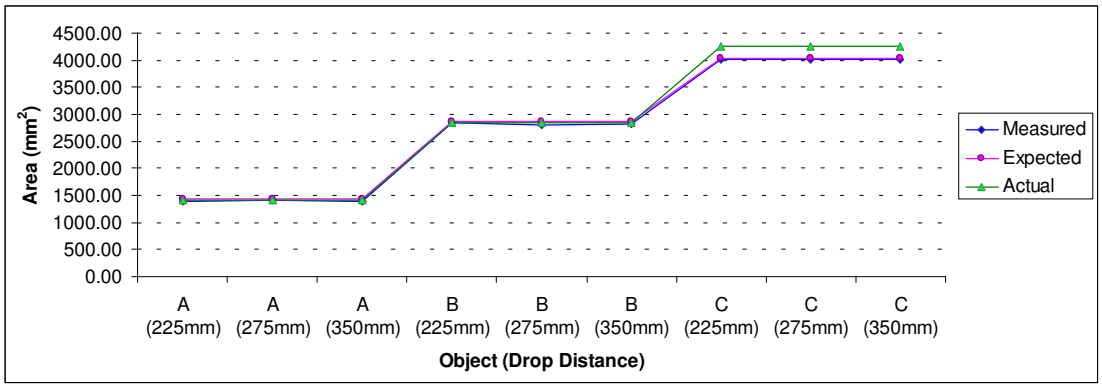


Figure 5.19 : Graphical Representation of Table 5.31

5.2.2.5 Discussion

From the results obtained, it can be seen that they have very small error (less than 2.5%) compared to the expected measurement reading as in during the Static Object image reconstruction experiments. This proves that the implemented system will be capable of producing good quality images during the object drop.

5.3 Flow velocity measurement

In this part, the results from the flow velocity measurements are presented. As discussed in Section 4.4.1 previously, cross correlation function was used to determine the time delay required by flowing material to pass through between two layers of sensors in a pipeline. Both the methods previously discussed (PTP correlation and STS correlation) to obtain flow velocity were applied and compared. Optimization using RFFT was also implemented.

The experiments were carried out with the drop distance and the objects cross sectional area were variables. In these experiments, the same three test objects used in previous experiment (image reconstruction) were used to simulate continuous solid flow. The experiment setting was also similar to the setting used in image reconstruction for dropping object. Each test object was dropped from three different drop distances (225 mm, 275 mm and 350 mm) and its velocity was measured. The drop distance was measured from the center of the object to the center of the sensor fixture. For all the experiments, 128 cross sectional images (hereafter referred as frames) for each top and bottom sensor layers were captured by the Data Acquisition Unit and stored by the Main Controller Unit in the external RAM. After data acquisition, the 256 data frames (128 Top layer, 128 Bottom layer) were used to reconstruct cross sectional images. This was followed by cross correlation analysis using pixel to pixel correlation to determine the flowing velocity. Then, the same data were analysed using sensor to sensor correlation for comparison. For all the experiments, the cross correlation algorithm used was the optimized Fourier domain cross correlation algorithm.

The following results are presented in the sequence of different drop distances and finally a combination of all results for comparison. In each table, 10 velocities measured using Object A, 10 velocities measured using Object B and 10 velocities measured using Object C are plotted. These data are plotted in a graph that follows immediately after each table. The velocities shown were actually averaged from the velocity profile obtained to ease analysis and comparison.

5.3.1 Measured Velocity from 225 mm drop distance

Table 5.32 : Measured velocity when drop distance is 225 mm

Flow	PTP	STS	Theory	Error $_{(STS-PTP)}$	Error $_{(STS-Theory)}$
25 %	1.8439	2.0306	2.2306	10.13%	-8.97%
	2.0271	2.2418	2.2306	10.59%	0.50%
	2.1584	2.0684	2.2306	-4.17%	-7.27%
	1.9835	2.1903	2.2306	10.43%	-1.81%
	2.0193	2.0300	2.2306	0.53%	-8.99%
	2.1950	2.0680	2.2306	-5.79%	-7.29%
	2.0788	2.0851	2.2306	0.30%	-6.52%
	1.8932	2.0694	2.2306	9.31%	-7.23%
	1.9055	2.1858	2.2306	14.71%	-2.01%
	2.0021	2.1417	2.2306	6.97%	-3.99%
50 %	1.9862	2.0504	2.2306	3.23%	-8.08%
	2.0607	2.1231	2.2306	3.03%	-4.82%
	1.9602	2.0624	2.2306	5.21%	-7.54%
	2.4437	2.0111	2.2306	-17.70%	-9.84%
	2.2566	2.0551	2.2306	-8.93%	-7.87%
	2.1594	2.0648	2.2306	-4.38%	-7.43%
	1.9847	1.9855	2.2306	0.04%	-10.99%
	2.0402	2.0236	2.2306	-0.81%	-9.28%
	2.0113	2.0252	2.2306	0.69%	-9.21%
	2.2969	2.1522	2.2306	-6.30%	-3.51%
75 %	1.9680	1.8414	2.2306	-6.43%	-17.45%
	1.9912	1.9575	2.2306	-1.69%	-12.24%
	2.0210	1.8442	2.2306	-8.75%	-17.32%
	1.9188	1.8760	2.2306	-2.23%	-15.90%
	1.9932	1.9757	2.2306	-0.88%	-11.43%
	2.1090	1.9094	2.2306	-9.46%	-14.40%
	2.2475	1.8652	2.2306	-17.01%	-16.38%
	1.9886	2.0368	2.2306	2.42%	-8.69%
	1.9338	2.0241	2.2306	4.67%	-9.26%
	2.2135	1.8725	2.2306	-15.41%	-16.05%

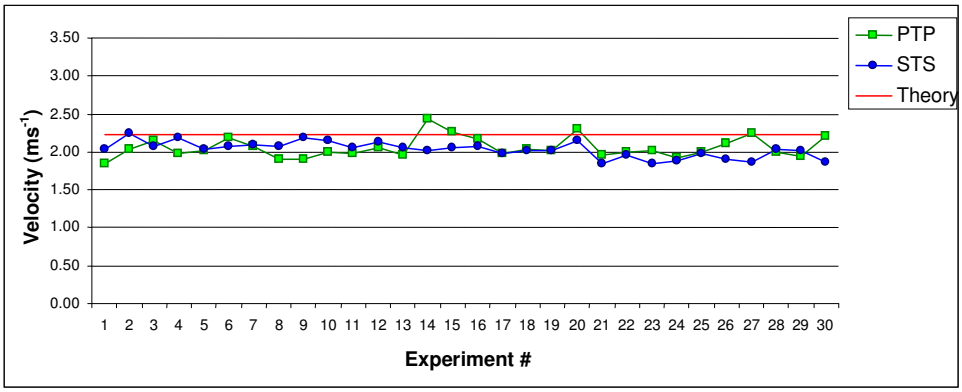


Figure 5.20 : Measured velocity from 225 mm drop distance

5.3.2 Measured velocity from 275 mm drop distance

Table 5.33 : Measured velocity when drop distance is 275 mm

Flow	PTP	STS	Theory	Error (STS-PTP)	Error (STS-Theory)
25 %	2.2064	2.3505	2.4512	6.53%	-4.11%
	2.1097	2.3664	2.4512	12.17%	-3.46%
	2.2218	2.3802	2.4512	7.13%	-2.90%
	2.1688	2.3631	2.4512	8.96%	-3.59%
	2.2395	2.3296	2.4512	4.02%	-4.96%
	2.2266	2.3476	2.4512	5.43%	-4.23%
	2.1411	2.3674	2.4512	10.57%	-3.42%
	2.3262	2.4057	2.4512	3.42%	-1.86%
	2.1626	2.5370	2.4512	17.31%	3.50%
	2.5516	2.3584	2.4512	-7.57%	-3.79%
50 %	2.2525	2.3668	2.4512	5.07%	-3.44%
	2.2453	2.2457	2.4512	0.02%	-8.38%
	2.2897	2.2581	2.4512	-1.38%	-7.88%
	2.2634	2.3094	2.4512	2.03%	-5.78%
	2.2481	2.3144	2.4512	2.95%	-5.58%
	2.5830	2.3182	2.4512	-10.25%	-5.43%
	2.1596	2.2840	2.4512	5.76%	-6.82%
	2.2298	2.3083	2.4512	3.52%	-5.83%
	2.4876	2.3858	2.4512	-4.09%	-2.67%
	2.3284	2.2723	2.4512	-2.41%	-7.30%
75 %	2.4276	2.2507	2.4512	-7.29%	-8.18%
	2.4307	2.3368	2.4512	-3.86%	-4.67%
	2.1035	2.2080	2.4512	4.97%	-9.92%
	2.2792	2.3102	2.4512	1.36%	-5.75%
	2.1509	2.1717	2.4512	0.97%	-11.40%
	2.1487	2.1734	2.4512	1.15%	-11.33%
	2.1214	2.1970	2.4512	3.56%	-10.37%
	2.1238	2.2360	2.4512	5.28%	-8.78%
	2.1732	2.2294	2.4512	2.59%	-9.05%
	2.3016	2.1829	2.4512	-5.16%	-10.95%

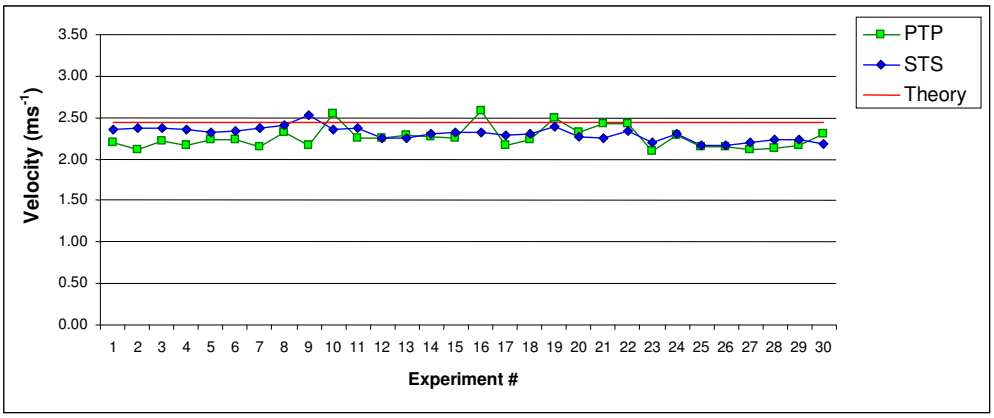


Figure 5.21 : Measured velocity from 275 mm drop distance

5.3.3 Measured Velocity from 350 mm drop distance

Table 5.34 : Measured velocity when drop distance is 350 mm

Flow	PTP	STS	Theory	Error (STS-PTP)	Error (STS-Theory)
25 %	2.5724	2.7951	2.7442	8.66%	1.85%
	2.6394	2.5977	2.7442	-1.58%	-5.34%
	2.4949	2.6255	2.7442	5.23%	-4.33%
	2.4069	2.8754	2.7442	19.46%	4.78%
	2.5751	2.7430	2.7442	6.52%	-0.04%
	2.4239	2.7028	2.7442	11.51%	-1.51%
	2.8380	2.6153	2.7442	-7.85%	-4.70%
	2.5019	2.6594	2.7442	6.30%	-3.09%
	2.5808	2.9356	2.7442	13.75%	6.97%
	2.4365	2.7453	2.7442	12.67%	0.04%
50 %	2.6809	2.5982	2.7442	-3.08%	-5.32%
	2.6836	2.8039	2.7442	4.48%	2.18%
	2.5415	2.6196	2.7442	3.07%	-4.54%
	2.4935	2.5329	2.7442	1.58%	-7.70%
	2.5688	2.5973	2.7442	1.11%	-5.35%
	2.5469	2.6308	2.7442	3.29%	-4.13%
	2.4549	2.5555	2.7442	4.10%	-6.88%
	2.4731	2.5625	2.7442	3.61%	-6.62%
	2.6553	2.6435	2.7442	-0.44%	-3.67%
	2.5728	2.7605	2.7442	7.30%	0.59%
75 %	2.8124	2.5650	2.7442	-8.80%	-6.53%
	2.7720	2.5968	2.7442	-6.32%	-5.37%
	2.4656	2.6115	2.7442	5.92%	-4.84%
	2.5937	2.6025	2.7442	0.34%	-5.16%
	2.9924	2.7388	2.7442	-8.47%	-0.20%
	2.9767	2.5316	2.7442	-14.95%	-7.75%
	2.4251	2.6047	2.7442	7.41%	-5.08%
	2.9311	2.7263	2.7442	-6.99%	-0.65%
	2.6302	2.5670	2.7442	-2.40%	-6.46%
	2.9725	2.6092	2.7442	-12.22%	-4.92%

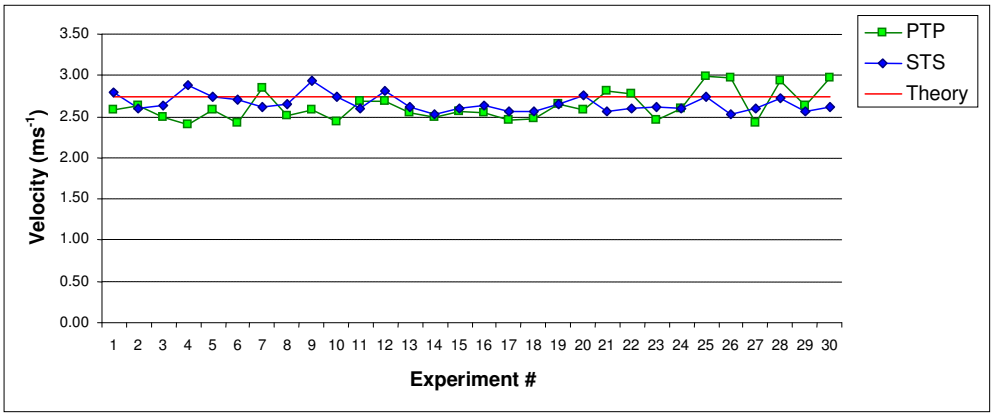


Figure 5.22 : Measured velocity for 350 mm drop distance

5.3.4 Summary of Measured Velocity Results

Shown in the following Table 5.35 is the average velocity calculated from the results obtained as in Table 5.32 to Table 5.34 using sensor to sensor cross correlation. This is followed by Table 5.36 which shows the theoretical velocity if the experiments were carried out in vacuum (where there was no air resistance presence). The theoretical velocities were calculated according to the following Equation 5.3 (Pang, 2004a).

$$v = \sqrt{2gh} \qquad \ldots 5.3$$

Where,

v = Theoretical object velocity

g = Gravity acceleration

h = Drop distance

Pang (2004a) used this equation to compare with experimental results he obtained using signal analyser instrument. He concluded that they were identical with only small error.

Figure 5.23 shows the plotted graph of the average velocity and the theoretical velocity.

Table 5.35 : Summary of average measured velocity using STS

Flow / Distance	25 %	50 %	75 %
225 mm	2.1111	2.0553	1.9203
275 mm	2.3806	2.3063	2.2296
350 mm	2.7295	2.6305	2.6153

Table 5.36 : Theoretical velocities

Flow / Distance	25 %	50 %	75 %
225 mm	2.2306	2.2306	2.2306
275 mm	2.4512	2.4512	2.4512
350 mm	2.7442	2.7442	2.7442

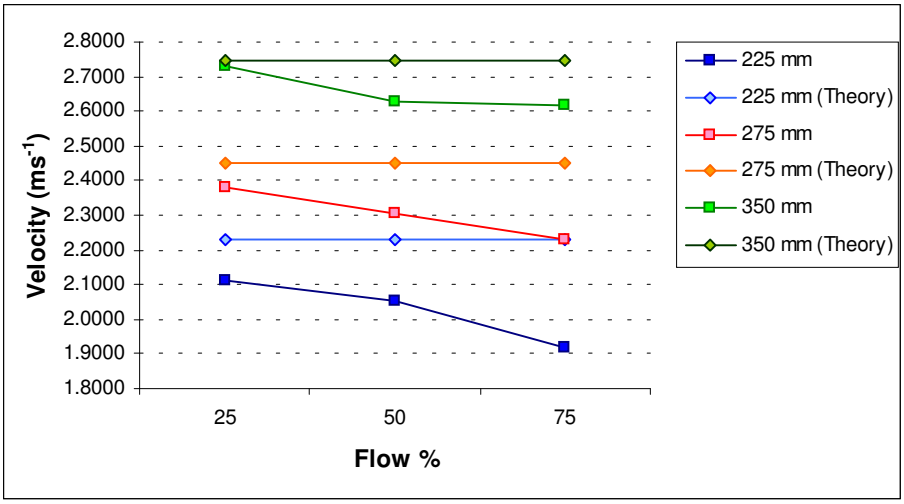


Figure 5.23 : Comparison of average and theoretical velocities

5.3.5 Discussion

From the results in Table 5.32 to Table 5.34, it can be seen that the STS correlation offers results very close to that from PTP correlation. The difference lies in the resulting velocity profile, where PTP correlation produces a profile with no smearing effect while the STS correlation produces otherwise. This smearing was due to the use of back projection technique onto the periphery velocities. Anyway, the smearing effect will be eliminated automatically during the calculation of the mass flow rate where the cross sectional images without smearing was multiplied pixel by pixel to the velocity profile.

In comparison with theoretical calculations, the error obtained was within acceptable range. The velocity obtained from theoretical calculations as shown in Table 5.36 assumed that air resistance was negligible. However, it was shown during the experiments and results as in Figure 5.23 where the velocities measured were actually affected by air resistance. It can be seen that as the cross sectional area of the flow becomes larger (25% → 50% → 75%), the velocity measured for the same drop distance becomes lower than theoretical values. As the cross sectional area increases, more air particles hit the object during free fall. This generates a force that pushes the object upwards (known as drag), causing the net velocity to be lower than calculated theoretically (as in free fall). This behaviour was identical for all the objects being used.

On the other hand, the velocities measured for the same object has smaller error percentage compared to theoretical calculations (for the same object comparison) as the drop distance increases. This behaviour was consistent for all the objects being used. This phenomenon can be explained by considering that the velocity during short drop distance was easily affected by drag (force upwards) due to air resistance. On the contrary, longer drop distance resulted in higher velocity when the object reaches the measurement level and therefore the drag caused by air resistance was negligible.

5.4 Mass Flow Rate Measurement

This part of results focuses on the Mass Flow Rate (MFR) measurement using the designed instrument. The experiments carried out used the same test objects (Object A, Object B and Object C) as used in image reconstruction and velocity measurement experiments. The physical properties of the test objects were as shown in Table 5.37. The setting for the experiments were similar to that of velocity measurement; where the test objects were dropped one at a time for MFR measurement.

Table 5.37 : Properties of test objects for MFR measurement

	Object A	**Object B**	**Object C**
Radius (mm)	21.25	30.05	36.81
Cross Sectional Area (mm^2)	1418.63	2837.25	4255.88
Height (mm)	120	120	120
Volume (mm^3)	170235.0519	340470.1038	510705.1557
Weight (g)	25	50	75
Density ($g\ mm^{-3}$)	0.000147	0.000147	0.000147

In the following, results are presented according to drop distance and finally a summary of the results are tabulated and plotted in a graph. Based on Table 5.37, the theoretical MFR can be calculated as follows:

$$V_u(x,y) = \sqrt{2gh_u} \qquad ...5.4$$

$$V_d(x,y) = \sqrt{2gh_d} \qquad ...5.5$$

$$V_{avg}(x,y) = \frac{V_u(x,y) + V_d(x,y)}{2} \qquad ...5.6$$

$$MFR\ (x,y) = A_{obj}\ (x,y)\ \text{x}\ V_{avg}(x,y)\ \text{x}\ \rho_{obj} \qquad ...5.7$$

Where,

$V_u(x,y)$ = Upstream velocity at pixel (x,y)

$V_d(x,y)$ = Downstream velocity at pixel (x,y)

g = Gravity acceleration

h_u = Distance of drop level to upstream sensor level

h_d = Distance of drop level to downstream sensor level

$V_{avg}(x, y)$ = Average velocity at pixel (x,y)

MFR (x,y) = MFR at pixel (x,y)

A_{obj} = Concentration value at pixel (x,y)

ρ_{obj} = Object density

5.4.1 MFR at 225 mm drop distance

Table 5.38 : MFR when drop distance is 225 mm

Object	MFR $_{(PTP)}$ (gs^{-1})	MFR$_{(STS)}$ (gs^{-1})	MFR $_{(theory)}$ (gs^{-1})	%Error $_{(STS-PTP)}$	%Error $_{(STS-theory)}$
A	405	445	465	9.88	-4.30
	417	440	465	5.52	-5.38
	417	434	465	4.08	-6.67
	418	473	465	13.16	1.72
	421	428	465	1.66	-7.96
	421	429	465	1.90	-7.74
	422	454	465	7.58	-2.37
	428	460	465	7.48	-1.08
	431	435	465	0.93	-6.45
	437	416	465	-4.81	-10.54
B	802	804	930	0.25	-13.55
	803	824	930	2.62	-11.40
	811	814	930	0.37	-12.47
	812	830	930	2.22	-10.75
	817	829	930	1.47	-10.86
	822	827	930	0.61	-11.08
	841	835	930	-0.71	-10.22
	865	810	930	-6.36	-12.90
	888	834	930	-6.08	-10.32
	895	852	930	-4.80	-8.39
C	1124	1118	1395	-0.53	-24.78
	1133	1174	1395	3.62	-18.82
	1153	1087	1395	-5.72	-28.33
	1187	1146	1395	-3.45	-21.73
	1207	1164	1395	-3.56	-19.85
	1208	1150	1395	-4.80	-21.30
	1229	1140	1395	-7.24	-22.37
	1232	1181	1395	-4.14	-18.12
	1318	1143	1395	-13.28	-22.05
	1355	1181	1395	-12.84	-18.12

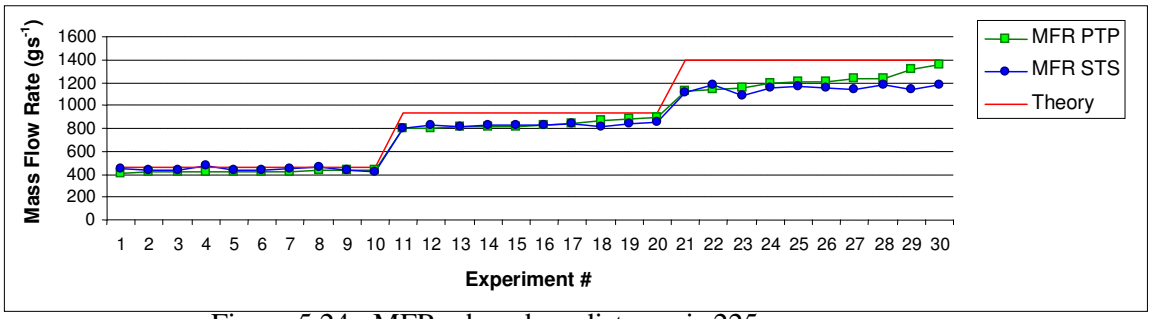


Figure 5.24 : MFR when drop distance is 225 mm

5.4.2 MFR at 275 mm drop distance

Table 5.39 : MFR when drop distance is 275 mm

Object	MFR (PTP) (gs^{-1})	MFR(STS) (gs^{-1})	MFR (theory) (gs^{-1})	Error (STS-PTP) (%)	Error (STS-theory) (%)
A	442	506	504	14.48	8.82
	453	492	504	8.61	5.81
	454	475	504	4.63	2.15
	457	496	504	8.53	6.67
	459	470	504	2.40	1.08
	460	492	504	6.96	5.81
	465	488	504	4.95	4.95
	470	518	504	10.21	11.40
	488	505	504	3.48	8.60
	534	510	504	-4.49	9.68
B	888	935	1007	5.29	-7.15
	899	923	1007	2.67	-8.34
	900	938	1007	4.22	-6.85
	909	912	1007	0.33	-9.43
	916	914	1007	-0.22	-9.24
	917	910	1007	-0.76	-9.63
	920	929	1007	0.98	-7.75
	938	952	1007	1.49	-5.46
	955	962	1007	0.73	-4.47
	965	895	1007	-7.25	-11.12
C	1268	1323	1510	4.34	-12.38
	1273	1300	1510	2.12	-13.91
	1278	1326	1510	3.76	-12.19
	1289	1349	1510	4.65	-10.66
	1299	1325	1510	2.00	-12.25
	1307	1357	1510	3.83	-10.13
	1323	1363	1510	3.02	-9.74
	1358	1303	1510	-4.05	-13.71
	1420	1330	1510	-6.34	-11.92
	1435	1375	1510	-4.18	-8.94

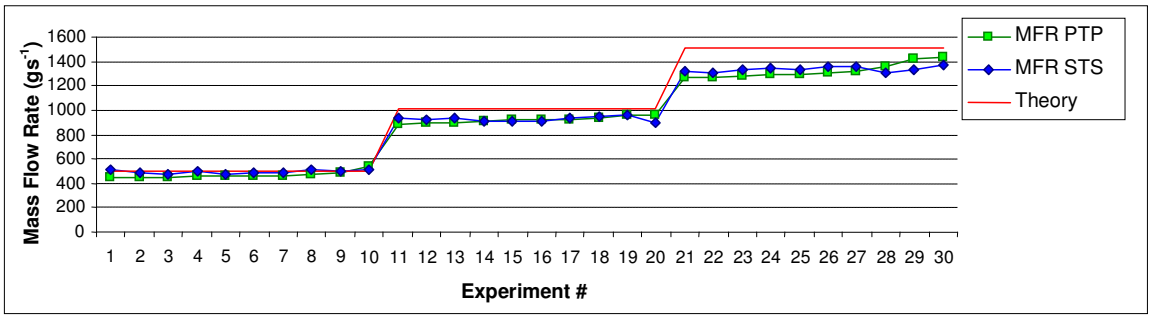


Figure 5.25 : MFR when drop distance is 275 mm

5.4.3 MFR at 350 mm drop distance

Table 5.40 : MFR when drop distance is 350 mm

Object	MFR (PTP) (gs^{-1})	MFR(STS) (gs^{-1})	MFR (theory) (gs^{-1})	Error (STS-PTP) (%)	Error (STS-theory) (%)
A	469	507	572	8.10	-11.36
	483	535	572	10.77	-6.47
	496	538	572	8.47	-5.94
	532	568	572	6.77	-0.70
	535	545	572	1.87	-4.72
	537	530	572	-1.30	-7.34
	546	628	572	15.02	9.79
	548	621	572	13.32	8.57
	560	597	572	6.61	4.37
	599	555	572	-7.35	-2.97
B	972	1044	1145	7.41	-8.82
	1004	1090	1145	8.57	-4.80
	1022	1074	1145	5.09	-6.20
	1027	1070	1145	4.19	-6.55
	1037	1055	1145	1.74	-7.86
	1041	1043	1145	0.19	-8.91
	1044	1087	1145	4.12	-5.07
	1050	1034	1145	-1.52	-9.69
	1062	1078	1145	1.51	-5.85
	1112	1156	1145	3.96	0.96
C	1452	1549	1717	6.68	-9.78
	1455	1558	1717	7.08	-9.26
	1538	1555	1717	1.11	-9.44
	1539	1526	1717	-0.84	-11.12
	1566	1578	1717	0.77	-8.10
	1598	1508	1717	-5.63	-12.17
	1635	1554	1717	-4.95	-9.49
	1691	1562	1717	-7.63	-9.03
	1742	1550	1717	-11.02	-9.73
	1837	1643	1717	-10.56	-4.31

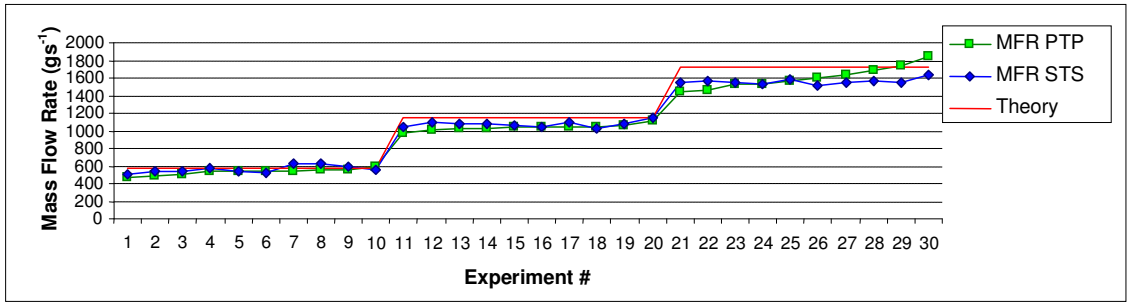


Figure 5.26 : MFR when drop distance is 350 mm

5.4.4 Summary of MFR measurement results

From the results obtained (Table 5.38 - Table 5.40), the MFR obtained were averaged and tabulated in the following Table 5.41. Table 5.42 shows the theoretical MFR results.

Table 5.41 : Summary of measured MFR results

Flow / Distance	25 % (gs^{-1})	50 % (gs^{-1})	75 % (gs^{-1})
225 mm	441.4	825.9	1148.4
275 mm	495.2	927	1335.1
350 mm	562.4	1073.1	1558.3

Table 5.42 : Theoretical MFR results

Flow / Distance	25 % (gs^{-1})	50 % (gs^{-1})	75 % (gs^{-1})
225 mm	465	930	1395
275 mm	504	1007	1510
350 mm	572	1145	1717

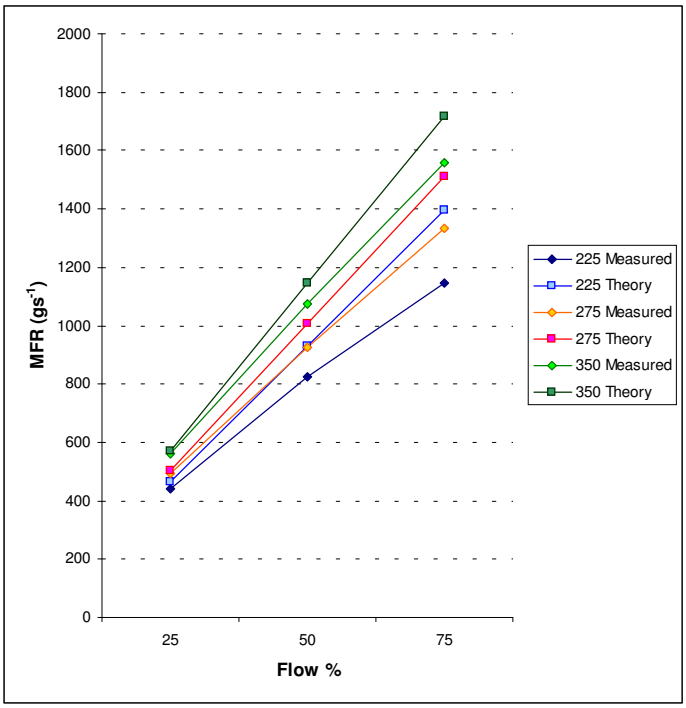


Figure 5.27 : Comparison of average and theoretical MFR

5.4.5 Discussion

The results obtained showed some correlations between measurements in image reconstruction experiments, velocity profile measurement and MFR measurement. For image reconstruction results, the measurements were quite consistent even for different flow conditions. However, the flow velocity measured was not as consistent due to the effects of air resistance. As a result, the MFR calculated was also affected in the same pattern. For example, the velocity measured was most accurate when the object cross sectional area was smallest and the drop distance was the longest affected the MFR to have similar result.

Overall, the MFR measurements performed using Object A (25%) for all drop distances resulted in error of less than 12% compared to theoretical calculations. On the contrary, the MFR error (compared to theoretical calculations) could reach up to almost 25% when Object C (75%) was being used and the drop distance was the shortest. However, this comparison is only based on the theoretical calculations

which ignore the effects of air resistance. Logically, this measurement makes sense as the larger cross sectional area causes higher air resistance which in turn reduces the velocity of the falling object and thus the MFR.

5.5 Real Time System Performance Benchmark

In this part of results, the performance of the implemented system was benchmarked. Basically, an oscilloscope was used to measure the speed of each tasks. On the DSP, a digital output was turned to HIGH to represent the start of an operation. After the operation completed, the digital output was turned to LOW. The period between the HIGH and LOW was captured and measured using the oscilloscope.

5.5.1 Data Acquisition Period

The data acquisition system is mainly controlled by the Main Controller Unit which was powered by TMS 320 F2812 Fixed point DSP. This system is capable of high speed sampling where only 390 micro seconds are needed to capture a frame of data (consist of 64 analogue measurements for Upstream and Downstream sensors). Figure 5.28 shows the scope capture of analogue data acquisition process.

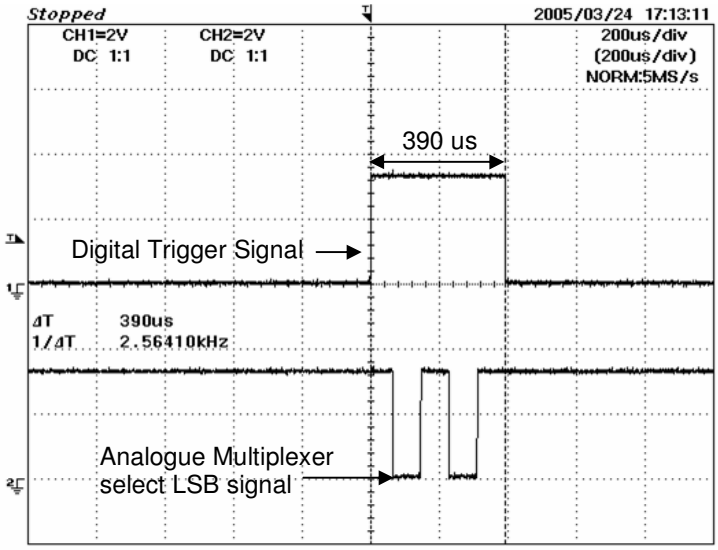


Figure 5.28 : Analogue data acquisition period

On the other hand, serial peripheral interface for the parallel to serial converter used in Digital Conditioning Unit was implemented at 9.17 Mbps. This is shown in Figure 5.29 where the serial clock signal was captured. As there were 92 binary bits going into the Main Controller Unit, 6 words (each word 16 bits) were transferred through the serial data line at each data acquisition frame. An example of data transfer is shown in Figure 5.30. Note that only 12.38 micro seconds was required to acquire all 92 digital bits from Upstream and Downstream rectilinear sensors.

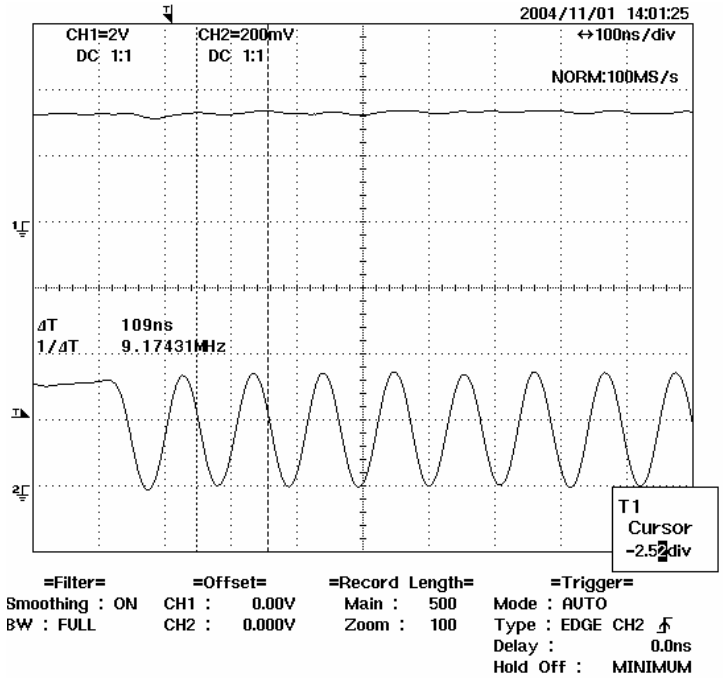


Figure 5.29 : Serial Peripheral Interface clock signal

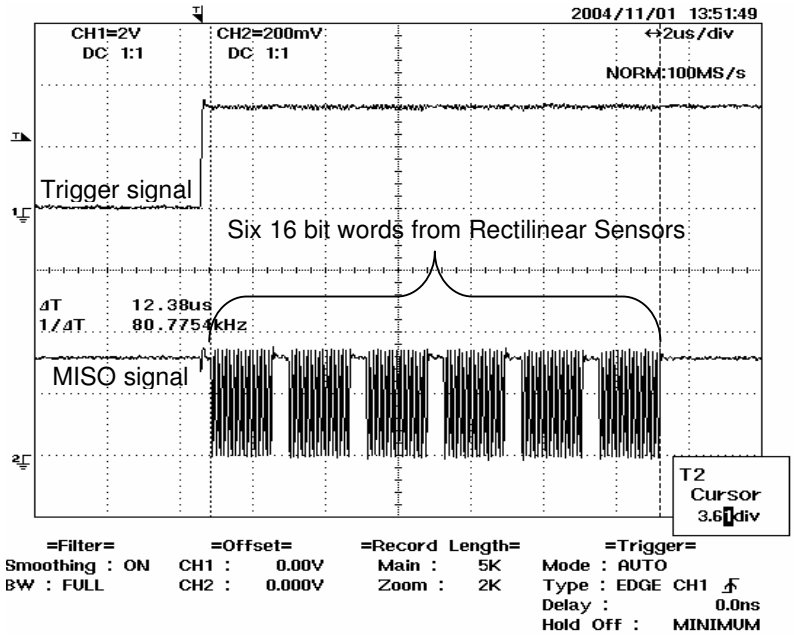


Figure 5.30 : Digital data acquisition period

On the other hand, the serial communication between the Main Controller Unit and PC GUI Unit using Serial Communication Interface was established at a

baud rate of 115200 bit per second (bps). This baud rate is the maximum achievable baud rate on a standard personal computer. At this baud rate, a period of 22.2 ms was required to transfer 256 bytes of data (a frame of reconstructed image), 3 bytes of program setting, 1 byte of a frame's period and 4 bytes for MFR result. The following Figure 5.31 shows an example of serial communication transfer captured using an oscilloscope.

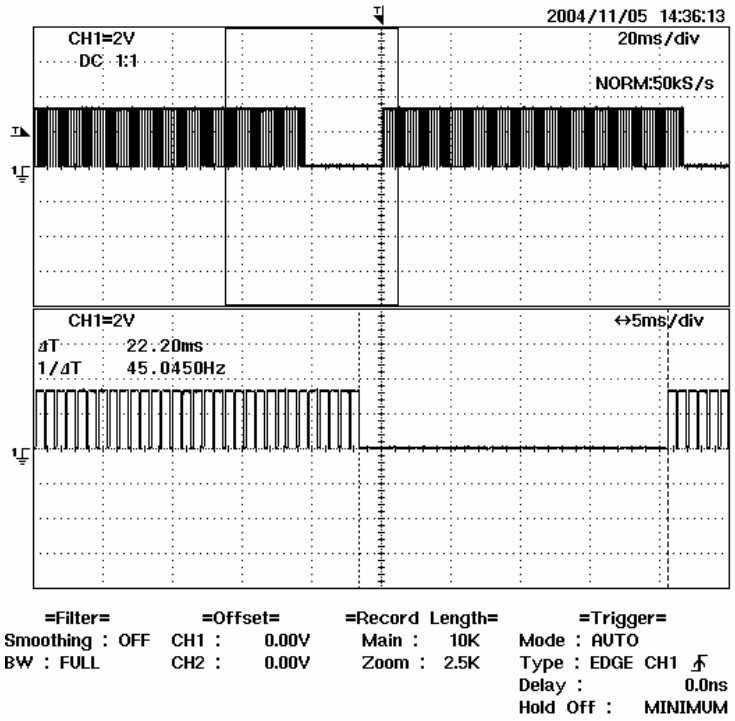


Figure 5.31 : Serial Communication Interface transfer time

5.5.2 Image Reconstruction Period

The image reconstruction algorithm being used in this research was the New Hybrid Reconstruction algorithm as discussed in Section 3.3 previously. Using this algorithm, the period of time required to perform image reconstruction of a single layer was 400 us. This is depicted in the following Figure 5.32. During MFR measurement, 128 frames of sensor data for each layer were captured. This means that there were 256 frames of sensor data to be processed. This process was very time consuming since it needed to access external memory for input reading (sensors data) and output storage (images reconstructed).

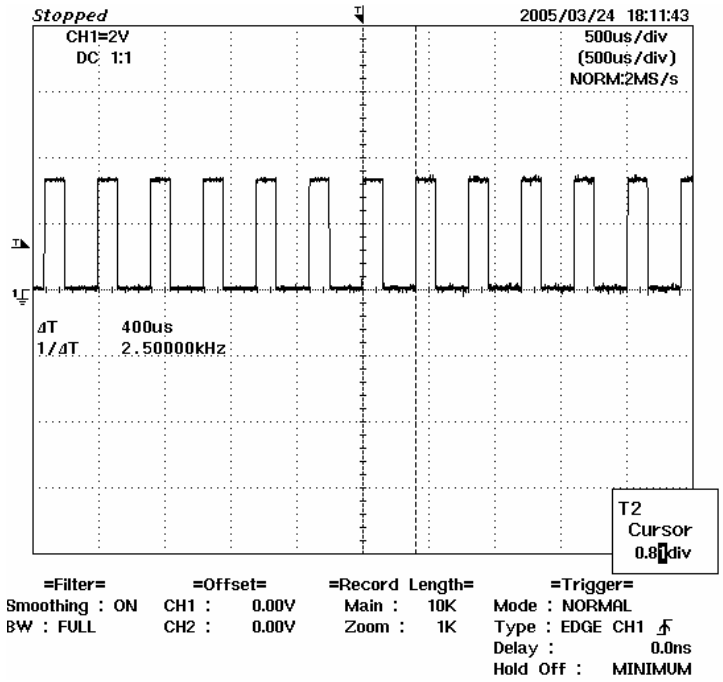


Figure 5.32 : Image reconstruction period

5.5.3 Velocity Profile Calculation Period

After reconstructing cross sectional images from the sensors data, velocity profile calculation was carried out. This process involved performing cross correlation on the acquired data. The reconstructed images were used as data for PTP correlation algorithm, whereas the sensors values were used as data for STS correlation algorithm. In terms of results, both algorithms perform with much similarity. However, the STS correlation algorithm worked much faster than the former.

The processing time shown in the following Table 5.43 shows time required for cross correlation function performed in time and frequency domain. Following that, Figure 5.33 shows the graph for this table.

Table 5.43 : Comparison between PTP and STS correlation in time domain and frequency domain.

Domain	Time		Frequency	
Number of Samples	64	128	64	128
PTP Correlation Period	194.29 ms	731.47 ms	206.61 ms	458.90 ms
STS Correlation Period	19.71 ms	72.13 ms	26.98 ms	59.56 ms

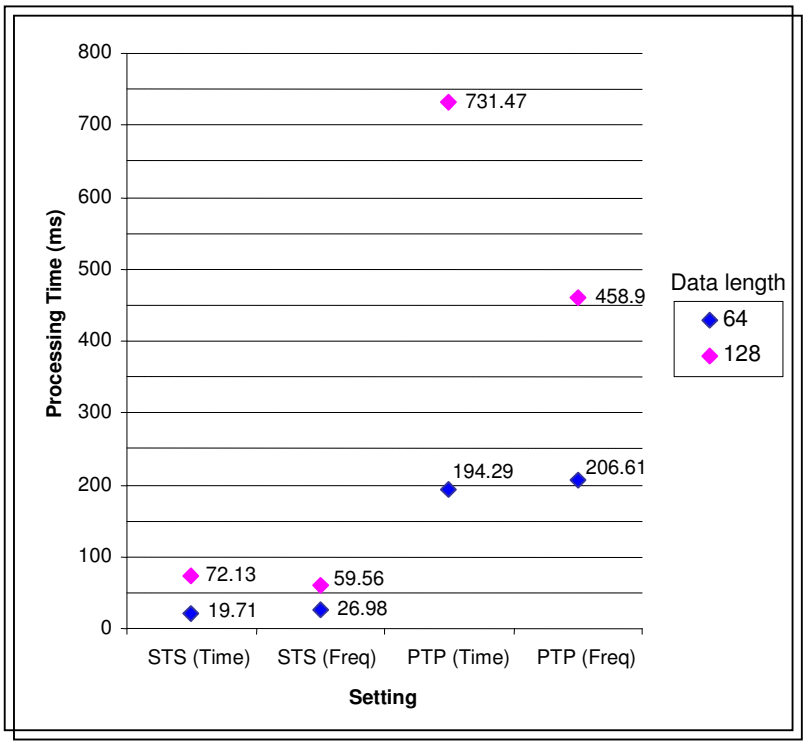


Figure 5.33 : Comparison of PTP and STS correlation processing time

From Figure 5.33, it can be calculated that the processing time required for cross correlation of 128 samples in time domain is about 3.6 times than the time required for 64 samples. On the contrary, the processing time required for 128 samples in frequency domain is only about 2.2 times than the time required for 64 samples. It can also be noted that the processing time for cross correlation in frequency domain is comparatively lower than required in time domain cross correlation when the number of samples is larger. As a result, Sensor to Sensor Correlation in frequency domain was used for all the velocity and mass flow rate measurements (shown in velocity profile results (Section 4.3) and mass flow rate results (Section 4.4)).

5.5.4 Mass Flow Rate Calculation Period

The mass flow rate processing time was calculated from the beginning of the completion of image reconstruction process until a mass flow rate result being generated. Therefore, this period included time required for cross correlation to generate velocity profile and also combining the results from image reconstruction, velocity profile and density of the solid object. An oscilloscope capture of the mass flow rate processing is shown in the following Figure 5.34. Only 174 ms was required for cross correlation and combination of all the information obtained from the previous processes before output a mass flow rate result. If we subtract the time required for STS correlation which was 59.56 ms, then the net period required to process a MFR result is only 114.44 ms.

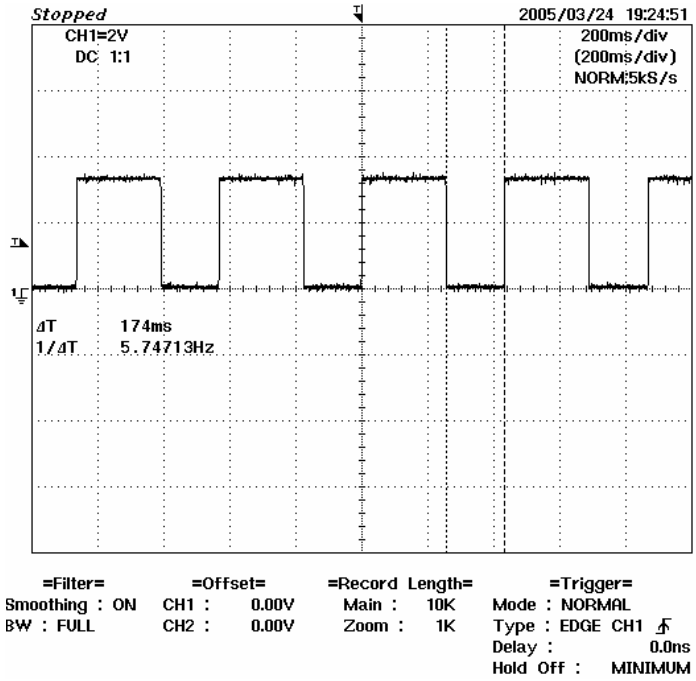


Figure 5.34 : Mass Flow Rate processing period

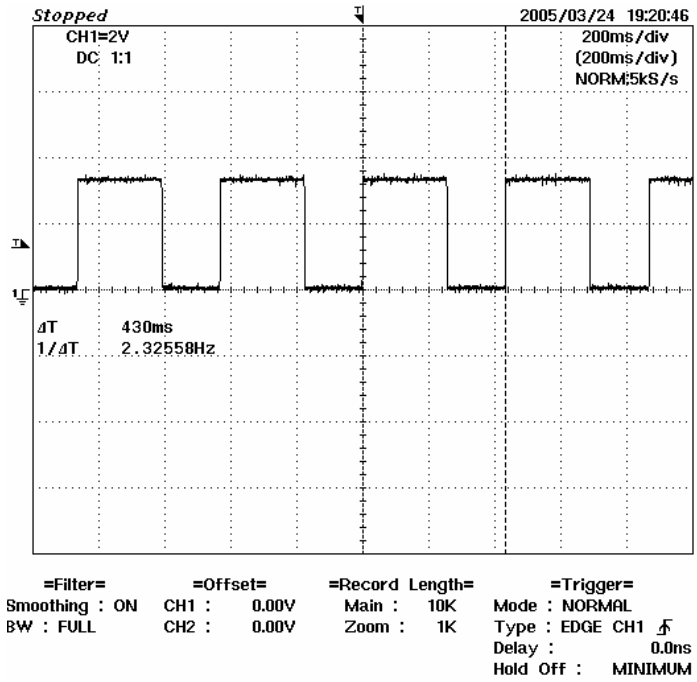


Figure 5.35 : Overall Mass Flow Rate processing period

In Figure 5.35, the oscilloscope capture shows that the overall time required to obtain a mass flow rate result is 430 ms. Actually, 256 ms (refer HIGH level in Figure 5.34 and compare with Figure 5.35) was used to perform data acquisition and image reconstruction processes. The rest of the time (174 ms) was used to perform cross correlation function (to obtain velocity profile), combine the results with concentration profile and calculate the mass flow rate. In comparison with work by Pang (2004d), 6 MFR can be calculated per second

The sampling rate in this research was set to 2 ms in order to capture longer continuous flow per mass flow rate result. Since only 128 sets of data were captured, a total of 256 ms period of flow was captured per mass flow rate measurement. If the sampling rate is higher, for example to 2 kHz; the total period captured will be only 64 ms. For this short duration, certain flow conditions may be difficult to be determined as the changes of flow signal in the limited samples available is not significant. As a result, cross correlation will not be able to return a valid result and flow velocity cannot be determined.

5.5.5 Discussion

The implementation of each part of the mass flow rate processing affects the overall performance of this system. Careful programming techniques were applied in order to produce highly optimized code. For example, long integer was used to represent decimal data instead of software based floating point format. This method is faster for a fixed point processor to execute as very little format conversion is required. However, there was limitation of the range of values that can be represented by the long integer compared to floating point format.

Another time saving feature being implemented was in terms of parameter passing. Passing a lot of parameters on a regularly called function wastes a lot of time since the parameters need to be copied to stack and when the function is being called repetitively, the overhead time accumulates to a noticeable amount. As a solution, global variables were used and only pointers to the global variables were passed to reference the data. Examples of such functions are image reconstruction function and cross correlation function.

Besides using pointers, using the internal memory of the DSP also greatly reduces processing time since the internal memory can be accessed without waiting states compared to external memory. For example, cross correlation function requires a lot of calculations which can be performed in internal memory and the results stored in external memory. This method is faster compared to keep on accessing external memory for each calculation steps.

Finally, the algorithm being used to implement the calculations is also important. This was shown previously in cross correlation results, where frequency domain algorithm is more efficient compared to time domain algorithm.

5.6 Summary

This chapter presented many results obtained from the research being carried out. The Data Acquisition System was the first part of the results where four PCB were fabricated for signal conditioning. The software of a DSP based embedded system to support and control the data acquisition process was also developed. The developed system is capable of capturing a set of data in only 390 us which consists of 64 analogue values from 4 orthogonal sensor layers and 92 digital bits from 4 rectilinear sensor layers.

Next, improvement was made to image reconstruction algorithm where the New Hybrid Reconstruction algorithm was proposed. The new algorithm provides more accurate measurement while maintaining the system performance. The new image reconstruction algorithm was tested with cross sectional area measurement for static object as well as dropping object. Results from the experiments showed that the new image reconstruction algorithm has error percentage of less than 5%.

Following image reconstruction, velocity measurement using optimization techniques of frequency domain processing and STS correlation instead of time domain processing and PTP correlation showed more significant reduction of processing time as the number of cross correlation increases. The proposed STS correlation was about 10 times faster than PTP correlation when both correlations were performed in time domain with 128 data elements. Similar comparison using frequency domain processing resulted in STS to perform about 7.7 times faster. In comparison between frequency domain processing and time domain processing, the former was about 1.6 times faster than the latter when using PTP correlation and was about 1.2 times faster when using STS correlation. The error percentage of the measured velocities compared to the theoretical calculations was less than 5% for 350 mm drop distance.

Results of Mass Flow Rate measurements were presented according to different drop distances and cross sectional objects. Using algorithm to calculate the mass flow rate, calibration constant was not required. Only density of the material

being used needed to be input to the system. The percentage of error measured was less than 2% for Object A (25% of full flow) at 350mm drop distance.

Finally, the overall performance of the system was benchmarked. The data acquisition took 256 ms to complete 128 sets of data to obtain a MFR result. This period was due to the sampling time of 2 ms per frame of data. As for the image reconstruction algorithm, it only required 400 us per sensor layer. The velocity profile calculation was basically the time required for the cross correlations and was affected by the choice of algorithm. For STS correlation of 128 frames of data and using frequency domain processing, only 59.56 ms were required. Finally, the mass flow rate calculation required 114.44 ms. The overall period required per MFR result was only 430 ms.

CHAPTER 6

CONCLUSIONS AND RECOMMENDATIONS

6.1 Conclusion

An online solid-gas Mass Flow Rate measurement system using optical tomography has been successfully developed. The specific objectives of the thesis have been fulfilled as below:

- A fixed point Digital Signal Processor (TMS320 F2812) has been used in the embedded system as the main controller for the MFR measurement system. This main controller has been successfully utilized to perform data acquisition, data analysis and sending result to other embedded system or computer host for display purpose. (Objective 1 met)

- High speed ADC being used has increased the data acquisition rate, where the period required to perform one set of measurements on analogue channels for the orthogonal layers has been reduced to only 390 micro seconds. For digital masking layer, hardware based comparators were successfully implemented to produce one bit digital value per sensor as an indication of the light beam being blocked or not. Using this technique, the serial communication between the sensor outputs and the main controller was established at 9.17 Mbps and only requires 12.38 us to complete a set of measurement. In total, each set of complete measurement only requires 402.38 us. (Objective 2 met)

- The New Hybrid Reconstruction algorithm used in this system requires only 400 micro seconds to complete reconstructing one cross sectional image. In terms of accuracy, the new algorithm reduces the possible errors in image reconstruction of arc shaped object using previous hybrid reconstruction algorithm by up to 12.5% per pixel. (Objective 3 met)

- Cross correlation function in frequency domain was successfully applied in this system to obtain flow velocity. Results obtained were compared to results from cross correlation in time domain and they were the same. However, frequency domain correlation outperforms time domain correlation about 1.6 times faster. (Objective 4 met)

- Sensor to Sensor correlation has been implemented in this system and the results obtained were compared to results from Pixel to Pixel correlation. From the comparison, the STS correlation can be used in the later stage for mass flow rate calculation. In terms of performance, the STS correlation outperforms PTP correlation at about 2.2 times faster (in frequency domain) and 3.7 times faster (in time domain). (Objective 5 met)

- The solid-gas mass flow rate firmware system has been successfully developed to run on a DSP powered embedded system to perform this mass flow rate measurement using algorithm-based and no other calibration constant.(Objective 6 met)

- The performance of image reconstruction, velocity profile measurement and mass flow rate measurements were captured on oscilloscope and presented. Overall, 430 ms is required to output a mass flow rate result. (Objective 7 met)

- Some suggestions and recommendations for future work are described in Section 5.3. (Objective 8 met)

There are a total of 5 major parts of results obtained and presented in this research. From Section 5.1, results from the data acquisition system were presented. Basically, major improvements were made in the performance of the data acquisition. This system was capable of acquiring a complete set of data in only 390 micro seconds or 2.56 kHz. Besides that, the hardware required to perform this task is only a small sized and easily portable embedded system.

From Section 5.2, image reconstruction has been carried out for 3 different cylindrical objects. The effects of different locations of static object and different drop distance have been investigated. The percentage of error for different locations during static object measurement was in the range of -4.95% to 0.98%. The percentage of error between the measured and expected results during object drop was all less than 2.5%.

From Section 5.3, different flow conditions were simulated by using 3 cylindrical objects and 3 different drop distances. The algorithm being used for these measurements was the improved frequency domain cross correlation function and 128 samples per velocity result. It can be concluded that longer drop distance and smaller cross sectional area resulted in more accurate velocity measurement compared to theoretical calculations. The effect of air resistance was noted during measurement of velocity using object with larger cross sectional area, where the measured velocity is lower than theoretical calculations (which neglect air resistance).

From Section 5.4, different conditions as in Section 4.3 were carried out to perform mass flow rate measurement. Since the results from image reconstruction were consistent for all conditions, it did not affect the mass flow rate measurement. On the other hand, the velocity being measured was affected by drop distance and cross sectional areas. The way it affected velocity measurement was seen in mass flow rate measurement with similar pattern. For Object A (25%), the maximum error percentage is less than 12% compared to expected value from theoretical calculations.

From Section 5.5, the performance of sub modules in the system was investigated. This involves data acquisition process, image reconstruction process, flow velocity measurement and mass flow rate measurement. On the other hand, research by Azrita (2002) required about 11.2 seconds to compute a mass flow rate result by using 156x32 (sampling signals x sensors) as a set of data in offline mode. This project improves her result significantly because only 0.43 second is needed to compute a mass flow rate result by using 100x156 (sampling signals x sensors) as a set of data in online mode.

6.2 Significant Contributions towards the body of knowledge in Process Tomography

The solid-gas mass flow rate meter based on embedded system proved that embedded system is capable of application in optical tomography system. The implementation of embedded system has the advantages of being portable, low cost and robust for industrial environment. Support to view the online concentration profile, velocity profile or mass flow rate using a laptop or standard personal computer with standard serial communication link further enhance the application of this system. Furthermore, the optimization techniques being used, namely the programming technique, data acquisition method and data analysis in frequency domain improved the performance of the overall system. There are still spaces for improvements and therefore it is hoped that a measurement instrument based on this system to be further developed and commercialised.

6.3 Recommendations for Future Work

i. This system exemplifies an optical tomography system running on a small size embedded system that is easily portable and requires no calibration prior to measurements. Further development can be carried out by using custom FPGA or ASIC for mass manufacturing. Another suitable device which is a recent breakthrough in technology is the Programmable System on Chip (PSoC). This device can be programmed with digital and analogue blocks to

support required peripherals such as ADC, SPI, UART and memory. This device also does not require other external support components as all necessary components are built in. The main benefit of using such device will be in much smaller, compact and higher performance system.

ii. Best correlated method (for pixel to pixel) or correlation of upstream sensors with several downstream sensors can be carried out to detect swirl flow. This method requires more processing power and possibly parallel processing architecture. A few embedded systems can be cascaded for this purpose.

iii. Multi-processor based embedded system can possibly further improve the performance of the overall system. Pipelining technology used in CPU design or parallel processing method can be implemented to increase the throughput.

iv. Higher throughput interface to PC GUI unit can be implemented by adding USB 2.0 technology. Using this technology, results from the flow meter can be sent to the PC at higher speed and thus, increases the frames display refresh rate.

v. All the PCB for Signal Conditioning Unit and Main Controller Unit can be designed together in a single PCB using surface mount technology. This will reduce the size of the overall system considerably.

REFERENCES

Azrita Binti Alias (2002). *Mass Flow Visualization Of Solid Particles In Pneumatic Pipelines Using Eletrodynamic Tomography System.* Universiti Teknologi Malaysia: M.Sc. Thesis.

Beck M S, Green R G and Thorn R (1987). Non-intrusive Measurement Of Solids Mass Flow In Pneumatic Conveying. *J. Phys. E: Sci. Instrum*. 20 No 7 (July 1987) 835-840

Jon Bates and Tim Tompkins (1998). *Using Visual C++ 6.* U.S.A: Que Corporation.

Byrne B., Coulthard J. and Yan Y. (1990). Radiological Sensors For Cross-correlation Flow Metering. *Nucleonic Instrumentation*, IEE Colloquium on 28 Feb 1990 Page(s):3/1 - 3/3

Chan Kok San (2002). *Real Time Image Reconstruction For Fan Beam Optical Tomography System*. Universiti Teknologi Malaysia: M.Sc. Thesis.

Christmann H.A., Smith D.R., Weaver B.L., Betten W.R. and Nazarian R.A. (1990). Real-time DSP System For Ultrasonic Blood Flow Measurement. Circuits and Systems, 1990., *IEEE International Symposium*. 1-3 May 1990 Page(s):2045 - 2048 vol.3

Dale Grover and John R. Deller (1999). *Digital Signal Processing and the microcontroller*. New Jersey: Prentice Hall Inc.

Deitel H. M. and Deitel P. J. (2001). *C++ How To Program.* Third Edition. New Jersey: Prentice Hall Inc.

Dickin F.J., Hoyle B.S., Hunt A., Huang S.M., Ilyas O., Lenn C., Waterfall R.C., Williams R.A., Xie C.G. and Beck M.S. (1992). Tomographic Imaging Of Industrial Process Equipment: Techniques And Applications. Circuits, Devices and Systems, *IEE Proceedings G*. Volume 139, Issue 1, Feb. 1992 Page(s):72 - 82

Dyakowski T (1996). Process Tomography Applied To Multi-phase Flow Measurement. *Meas. Sci. Technol*. 7 No 3 (March 1996) 343-353

Floyd Thomas L (1997). *Digital Fundamentals*. Sixth Edition. New Jersey: Prentice Hall Inc.

Foster S.G., Embree P.M. and O'Brien and W.D. Jr. (1990). Flow Velocity Profile Via Time-domain Correlation: Error Analysis And Computer Simulation Ultrasonics, Ferroelectrics and Frequency Control, *IEEE Transactions*. Volume 37, Issue 3, May 1990 Page(s):164 - 175

Green R G, Kwan H K, John R and Beck M S (1978). A Low-cost Solids Flowmeter For Industrial Use. *J. Phys. E: Sci. Instrum*. 11 No 10 (October 1978) 1005-1010

Green R G, Rahmat M F, Dutton K, Evans K, Goude A and Henry M (1997). Velocity And Mass Flow Rate Profiles Of Dry Powders In A Gravity Drop Conveyor Using An Electrodynamic Tomography System. *Meas. Sci. Technol*. 8 No 4 (April 1997) 429-436

Guilbert A.R. and Sanderson M.L. (1996). A Novel Ultrasonic Mass Flowmeter For Liquids. Advances In Sensors For Fluid Flow Measurement, *IEE Colloquium.* 18 Apr 1996 Page(s):8/1 - 8/4

Ibrahim S., Green R G, Dutton K, Evans K, Rahim R A and Goude A (1999). Optical Sensor Configurations For Process Tomography. *Meas. Sci. Technol.* Issue 11. November 1999.

Ibrahim S., Green R G, Dutton K and Rahim R A (2000). Modelling To Optimize The Design Of Optical Tomography Systems For Process Measurement. *Symposium On Process Tomography*. Jurata. 18-19 May 2000.

Ibrahim S, Green R G, Evans K, Dutton K and Rahim R A (2000). Optical Tomography For Process Measurement And Control. Control 2000. *UKACC Int Conference*. University of Cambridge. 4-7 Sept 2000.

Ibrahim S. and Green R.G. (2002a). Velocity Measurement Using Optical Sensors. Semiconductor Electronics, 2002. *Proceedings ICSE 2002*. IEEE International Conference on 19-21 Dec. 2002 Page(s):126 - 129

Ibrahim S. and Green R.G. (2002b). Optical Fibre Sensors For Imaging Concentration Profile. Semiconductor Electronics, 2002. Proceedings. *ICSE 2002*. IEEE International Conference on 19-21 Dec. 2002 Page(s):121 - 125

Kak A. C. and Slaney Malcolm (1999). *Principles Of Computerized Tomographic Imaging*. New York: IEEE Press.

Kenneth A Reek (1998). *Pointers On C*. U.S.A: Longman Inc.

Leong Lai Chen (2004). *Implementation Of Multiple Fan Beam Projection Technique In Optical Fibre Process Tomography*. Universiti Teknologi Malaysia: M. Sc. Thesis.

Mohd Hafiz Fazalul Rahiman (2002). *Image Reconstruction Using Transmission-mode of Ultrasonic Tomography in Water-Particles Flow*. Universiti Teknologi Malaysia: B. Sc. Thesis.

Mohd. Hezri Fazalul Rahiman (2002). *Real Time Velocity Profile Generation of Powder Conveying Using Electrical Charge Tomography*. Universiti Teknologi Malaysia: M Sc. Thesis

Motorola (2001), *M68HC11 E Series Programming Reference Guide*. U.S.A.: Motorola Incorporated.

Oppenheim Alan V., Schafer Ronald W. and Buck John R. (1999). *Discrete-time Signal Processing*. New Jersey: Prentice Hall Inc.

Pang Jon Fea (2004a). *Real-time Velocity And Mass Flow Rate Measurement Using Infrared Tomography*. Universiti Teknologi Malaysia: M. Sc. Thesis.

Pang Jon Fea, Ruzairi Abdul Rahim, Chan Kok San (2004b). Real-time Image Reconstruction System Using Two Data Processing Units In Optical Tomography. *3rd World Congress On Industrial Process Tomography*. Banff, Canada:

Pang Jon Fea, Ruzairi Abdul Rahim, Chan Kok San (2004c). Real-time Velocity Profile Measurement Using Pixel-To-Pixel Velocity Method In Optical Tomography Instrument. *3rd World Congress On Industrial Process Tomography*. Banff, Canada:

Pang Jon Fea, Ruzairi Abdul Rahim, Chan Kok San (2004d). Real-time Mass Flow Rate In Solid-Gas Measurement Using Optical Tomography Technique. *3rd World Congress On Industrial Process Tomography*. Banff, Canada:

Pang Jon Fea, Ruzairi Abdul Rahim, Chan Kok San (2004e). Infrared Tomography Sensor Configuration Using 4 Parallel Beam Projecions. *3rd World Congress On Industrial Process Tomography*. Banff, Canada:

Philip Denbigh (1998). *System Analysis & Signal Processing With Emphasis On The Use Of MATLAB*. U.S.A: Longman Inc.

Rahim R A, Green R G, Horbury N, Dickin F J, Naylor B D and Pridmore T P (1996). Further Development Of A Tomographic Imaging System Using Optical Fibres For Pneumatic Conveyors. *Meas. Sci. Technol.* Vol 7 No 3 (March 1996) 419-422

Rahim R. A. and Green R G (1997). "A Review of Tomography Sensors and Applications." *Journal Electrika*.

RS (2001). 287-0536: *Tranma Alphanumeric Dot Matrix Liquid Crystal Displays With Backlighting*. RS Components

Schaefer A.W., Reynolds J.J., Marks D.L. and Boppart S.A. (2004). Real-time Digital Signal Processing-based Optical Coherence Tomography And Doppler Optical Coherence Tomography. *Biomedical Engineering*, IEEE Transactions on Volume 51, Issue 1, Jan 2004 Page(s):186 - 190

Texas Instruments (2002a). *SPRU051:TMS320F28x Serial Communication Interface (SCI) Peripheral Reference Guide*. Dallas, Texas: Texas Instruments Incorporated.

Texas Instruments (2002b). *SPRU059:TMS320F28x Serial Peripheral Interface (SPI) Peripheral Reference Guide*. Dallas, Texas: Texas Instruments Incorporated.

Texas Instruments (2002c). *SPRU060:TMS320F28x Analog-to-Digital Converter (ADC) Peripheral Reference Guide*. Dallas, Texas: Texas Instruments Incorporated.

Texas Instruments (2002d). *SPRU065:TMS320F28x Event Manager (EV) Peripheral Reference Guide*. Dallas, Texas: Texas Instruments Incorporated.

Texas Instruments (2002e). *SPRU067:TMS320F28x External Interface (XINTF) Peripheral Reference Guide*. Dallas, Texas: Texas Instruments Incorporated.

Texas Instruments (2002f). *SPRU078:TMS320F28x System Control and Interrupts Peripheral Reference Guide*. Dallas, Texas: Texas Instruments Incorporated.

Texas Instruments (2002g). *SPRU095:TMS320F28x Boot ROM Peripheral Reference Guide*. Dallas, Texas: Texas Instruments Incorporated.

Texas Instruments (2002h). *FFT Library: Module User's Guide C28x Foundation Software*. Dallas, Texas: Texas Instruments Incorporated.

Texas Instruments (2002i). *DSP Selection Guide*. Dallas, Texas: Texas Instruments Incorporated.

Texas Instruments (2002j). *SPRS174H: TMS320F2810, TMS320F2812 Digital Signal Processors Data Manual*. Dallas, Texas: Texas Instruments Incorporated.

Thiam C K, Cong T Z, Fea P J, Rahim R A, Loon G C (2004). Digital Sensors for Masking Purpose in Parallel Beam Optical Tomography System. *3rd International Symposium on Process Tomography*. September 9-10. Poland:158-160

Thorn R, Beck M S and Green R G (1982). Non-intrusive Methods Of Velocity Measurement In Pneumatic Conveying. *J. Phys. E: Sci. Instrum.* 15 1131-1139. Print publication: Issue 11 (November 1982)

Xie C G, Stott A L, Huang S M, Plaskowski A and Beck M S (1989). Mass-flow Measurement Of Solids Using Electrodynamic And Capacitance Transducers. *J. Phys. E: Sci. Instrum*. 22 No 9 (September 1989) 712-719

Yan Y. (1996). Mass Flow Measurement Of Bulk Solids In Pneumatic Pipelines. *Meas. Sci. Technol*. 7 No 12 (December 1996) 1687-1706

Yuan Peng and Yu Long Mo (2003). An Electrical Impedance Tomography System. Biomedical Engineering, 2003. *IEEE EMBS Asian-Pacific Conference.* 20-22 Oct. 2003 Page(s):280 - 281

Zbigniew Rucki and Zdzislaw Szczepanik (2000). Impedance Tomograpy - Measurement Problems. *Symposium On Process Tomography*. Jurata. 18-19 May 2000.

APPENDICES

APPENDIX A

Program listing of Data Acquisition task

```
for (gFrame=0; gFrame<N; gFrame++) {            // sample N sets of data
      ////////////////////////////////////////////////////////////////////////////
      // B4 is Parallel Load
      // Begin Rectilinear Data Acquisition
      GpioDataRegs.GPBCLEAR.bit.GPIOB4 = 1; // USE CLEAR
      for (i=0; i<10; i++) {};      // delay to allow parallel to serial conversion
      GpioDataRegs.GPBSET.bit.GPIOB4 = 1;   // PL = HI (LOW TO HIGH TRANS)

      // Read 96 data, each time 16 bits
      for (i=0; i<6; i++) {
            SpiaRegs.SPITXBUF=0xAAAA;              // dummy data to initiate SPI transfer
            // Wait until data is received
            SpiaRegs.SPIRXBUF = 0;
            while(SpiaRegs.SPISTS.bit.INT_FLAG !=1) { }
            rect[i] = SpiaRegs.SPIRXBUF;
      } // End Rectilinear Data Acquisition

      j=0;
      for (i=0; i<23; i++) { // 23 QU
            if ( (rect[i/16] & bit[j]) != 0)      // any of the bit is SET means it's TRUE
                  gSensQU[gFrame][22-i] = 0x80;
            else
                  gSensQU[gFrame][22-i] = 0;

            j++;
            if (j==16) j=0;
      }

      for (i=0; i<23; i++) { // 23 PU
            if ( (rect[(i+23)/16] & bit[j]) != 0)      // any of the bit is SET means it's TRUE
                  gSensPU[gFrame][22-i] = 0x80;
            else
                  gSensPU[gFrame][22-i] = 0;

            j++;
            if (j==16) j=0;
      }
```

```
j=0;
for (i=0; i<23; i++) { // 23 QD
	if ( (rect[(i+48)/16] & bit[j]) != 0)	// any of the bit is SET means it's TRUE
		gSensQD[gFrame][22-i] = 0x80;
	else
		gSensQD[gFrame][22-i] = 0;

	j++;
	if (j==16) j=0;
}

for (i=0; i<23; i++) { // 23 PD
	if ( (rect[(i+71)/16] & bit[j]) != 0)	// any of the bit is SET means it's TRUE
		gSensPD[gFrame][i] = 0x80;
	else
		gSensPD[gFrame][i] = 0;

	j++;
	if (j==16) j=0;
}
////////////////////////////////////////////////////////////////////////////////// end Rectilinear Acquisition

// @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
//  Orthogonal Sensors Acquisition (Up & Down)
// @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
// Run 4 loops, each loop taking in 16 values.
// A0..A7, B0..B7 ==> 1 Loop    A = ATAS, B = BAWAH
//
// Mux | B0 B1 B2 B3 B4 B5 B6 B7
// B A | A0 A1 A2 A3 A4 A5 A6 A7
// ------------------------------
// 0 0 : 00 01 02 03 04 05 06 07  => Loop 1
// 0 1 : 08 09 10 11 12 13 14 15  => Loop 2
// 1 0 : 16 17 18 19 20 21 22 23  => Loop 3
// 1 1 : 24 25 26 27 28 29 30 31  => Loop 4
//

for (i=0; i<4; i++) {

	// Reset to return to CONV00
	AdcRegs.ADCTRL2.bit.RST_SEQ1 = 1;   // Every 16 times / Loop

	// Set Analog Mux Selector
	switch (i) {
	case 0:
		GpioDataRegs.GPBCLEAR.bit.GPIOB0=1;
		GpioDataRegs.GPBCLEAR.bit.GPIOB1=1;
		break;
	case 1:
		GpioDataRegs.GPBSET.bit.GPIOB0=1;
		break;
	case 2:
		GpioDataRegs.GPBCLEAR.bit.GPIOB0=1;
		GpioDataRegs.GPBSET.bit.GPIOB1 =1;
		break;
	case 3:
		GpioDataRegs.GPBSET.bit.GPIOB0 =1;
		break;
	}
```

```
Delay(MUX_DELAY);

AdcRegs.ADCTRL2.all = 0x2000;                    // Trigger Start SEQ
while (AdcRegs.ADCST.bit.INT_SEQ1 == 0) {}  // Wait for interrupt
// Software wait = (HISPCP*2) * (ADCCLKPS*2) * (CPS+1) cycles
//               = (3*2)      * (1*2)        * (0+1)   = 12 cycles
asm(" RPT #11 || NOP");                          // Delay

AdcRegs.ADCST.bit.INT_SEQ1_CLR = 1;       // Clear the Flag

// Save the results (re-mapped by oChan[])
if (i==0) {
      // Y-axis TOP
      gSensMU[gFrame][15] =((AdcRegs.ADCRESULT0>>SHIFTADC));
      gSensMU[gFrame][14] =((AdcRegs.ADCRESULT1>>SHIFTADC));
      gSensMU[gFrame][13] =((AdcRegs.ADCRESULT2>>SHIFTADC));
      gSensMU[gFrame][12] =((AdcRegs.ADCRESULT3>>SHIFTADC));
      gSensMU[gFrame][11] =((AdcRegs.ADCRESULT4>>SHIFTADC));
      gSensMU[gFrame][10] =((AdcRegs.ADCRESULT5>>SHIFTADC));
      gSensMU[gFrame][9]  =((AdcRegs.ADCRESULT6>>SHIFTADC));
      gSensMU[gFrame][8]  =((AdcRegs.ADCRESULT7>>SHIFTADC));

      // Y-axis BOTTOM
      gSensMD[gFrame][15] =((AdcRegs.ADCRESULT8>>SHIFTADC));
      gSensMD[gFrame][14] =((AdcRegs.ADCRESULT9>>SHIFTADC));
      gSensMD[gFrame][13] =((AdcRegs.ADCRESULT10>>SHIFTADC));
      gSensMD[gFrame][12] =((AdcRegs.ADCRESULT11>>SHIFTADC));
      gSensMD[gFrame][11] =((AdcRegs.ADCRESULT12>>SHIFTADC));
      gSensMD[gFrame][10] =((AdcRegs.ADCRESULT13>>SHIFTADC));
      gSensMD[gFrame][9]  =((AdcRegs.ADCRESULT14>>SHIFTADC));
      gSensMD[gFrame][8]  =((AdcRegs.ADCRESULT15>>SHIFTADC));

}
else if (i==1) {
      // Y-axis TOP
      gSensMU[gFrame][7] =((AdcRegs.ADCRESULT0>>SHIFTADC));
      gSensMU[gFrame][6] =((AdcRegs.ADCRESULT1>>SHIFTADC));
      gSensMU[gFrame][5] =((AdcRegs.ADCRESULT2>>SHIFTADC));
      gSensMU[gFrame][4] =((AdcRegs.ADCRESULT3>>SHIFTADC));
      gSensMU[gFrame][3] =((AdcRegs.ADCRESULT4>>SHIFTADC));
      gSensMU[gFrame][2] =((AdcRegs.ADCRESULT5>>SHIFTADC));
      gSensMU[gFrame][1] =((AdcRegs.ADCRESULT6>>SHIFTADC));
      gSensMU[gFrame][0] =((AdcRegs.ADCRESULT7>>SHIFTADC));

      // Y-axis BOTTOM
      gSensMD[gFrame][7] =((AdcRegs.ADCRESULT8>>SHIFTADC));
      gSensMD[gFrame][6] =((AdcRegs.ADCRESULT9>>SHIFTADC));
      gSensMD[gFrame][5] =((AdcRegs.ADCRESULT10>>SHIFTADC));
      gSensMD[gFrame][4] =((AdcRegs.ADCRESULT11>>SHIFTADC));
      gSensMD[gFrame][3] =((AdcRegs.ADCRESULT12>>SHIFTADC));
      gSensMD[gFrame][2] =((AdcRegs.ADCRESULT13>>SHIFTADC));
      gSensMD[gFrame][1] =((AdcRegs.ADCRESULT14>>SHIFTADC));
      gSensMD[gFrame][0] =((AdcRegs.ADCRESULT15>>SHIFTADC));

}
else if (i==2) {
      // X-axis TOP
      gSensNU[gFrame][0] =((AdcRegs.ADCRESULT0>>SHIFTADC));
      gSensNU[gFrame][1] =((AdcRegs.ADCRESULT1>>SHIFTADC));
      gSensNU[gFrame][2] =((AdcRegs.ADCRESULT2>>SHIFTADC));
```

```
                gSensNU[gFrame][3] =((AdcRegs.ADCRESULT3>>SHIFTADC));
                gSensNU[gFrame][4] =((AdcRegs.ADCRESULT4>>SHIFTADC));
                gSensNU[gFrame][5] =((AdcRegs.ADCRESULT5>>SHIFTADC));
                gSensNU[gFrame][6] =((AdcRegs.ADCRESULT6>>SHIFTADC));
                gSensNU[gFrame][7] =((AdcRegs.ADCRESULT7>>SHIFTADC));

                // X-axis BOTTOM
                gSensND[gFrame][0] =((AdcRegs.ADCRESULT8>>SHIFTADC));
                gSensND[gFrame][1] =((AdcRegs.ADCRESULT9>>SHIFTADC));
                gSensND[gFrame][2] =((AdcRegs.ADCRESULT10>>SHIFTADC));
                gSensND[gFrame][3] =((AdcRegs.ADCRESULT11>>SHIFTADC));
                gSensND[gFrame][4] =((AdcRegs.ADCRESULT12>>SHIFTADC));
                gSensND[gFrame][5] =((AdcRegs.ADCRESULT13>>SHIFTADC));
                gSensND[gFrame][6] =((AdcRegs.ADCRESULT14>>SHIFTADC));
                gSensND[gFrame][7] =((AdcRegs.ADCRESULT15>>SHIFTADC));
            }
            else {
                // X-axis TOP
                gSensNU[gFrame][8] =((AdcRegs.ADCRESULT0>>SHIFTADC));
                gSensNU[gFrame][9] =((AdcRegs.ADCRESULT1>>SHIFTADC));
                gSensNU[gFrame][10] =((AdcRegs.ADCRESULT2>>SHIFTADC));
                gSensNU[gFrame][11] =((AdcRegs.ADCRESULT3>>SHIFTADC));
                gSensNU[gFrame][12] =((AdcRegs.ADCRESULT4>>SHIFTADC));
                gSensNU[gFrame][13] =((AdcRegs.ADCRESULT5>>SHIFTADC));
                gSensNU[gFrame][14] =((AdcRegs.ADCRESULT6>>SHIFTADC));
                gSensNU[gFrame][15] =((AdcRegs.ADCRESULT7>>SHIFTADC));

                // X-axis BOTTOM
                gSensND[gFrame][8] =((AdcRegs.ADCRESULT8>>SHIFTADC));
                gSensND[gFrame][9] =((AdcRegs.ADCRESULT9>>SHIFTADC));
                gSensND[gFrame][10] =((AdcRegs.ADCRESULT10>>SHIFTADC));
                gSensND[gFrame][11] =((AdcRegs.ADCRESULT11>>SHIFTADC));
                gSensND[gFrame][12] =((AdcRegs.ADCRESULT12>>SHIFTADC));
                gSensND[gFrame][13] =((AdcRegs.ADCRESULT13>>SHIFTADC));
                gSensND[gFrame][14] =((AdcRegs.ADCRESULT14>>SHIFTADC));
                gSensND[gFrame][15] =((AdcRegs.ADCRESULT15>>SHIFTADC));
            }

        } // End 4 x loop

        // Calibration
        if (isCalibrate) {
            for (i=0; i<16; i++) {
                gOffNU[i] = gSensNU[0][i];
                gOffND[i] = gSensND[0][i];
                gOffMU[i] = gSensMU[0][i];
                gOffMD[i] = gSensMD[0][i];
            }
            isCalibrate = 0; // dont calibrate after first pass
        }

        for (i=0; i<16; i++) {
            // 170 = 2/3 * 255
            gSensNU[gFrame][i] = gSensNU[gFrame][i] *128 / SENSORMAXVALUE;
            gSensND[gFrame][i] = gSensND[gFrame][i] *128 / SENSORMAXVALUE;
            gSensMU[gFrame][i] = gSensMU[gFrame][i] *128 / SENSORMAXVALUE;
            gSensMD[gFrame][i] = gSensMD[gFrame][i] *128 / SENSORMAXVALUE;
        }
}
```

APPENDIX B

Program listing of Image Reconstruction Algorithm

```
#define OFFSET_P 11
#define OFFSET_Q 0-4

void DrawImage(int nFrame, Uint16 layer)
{
	Uint16 y,x,index;
	int iQ, iP;

	register int *ortho, *sensM, *sensN, *sensP, *sensQ;
	Uint16 gSumCon, *p_gSumCon;
	Uint16 triangleCount;

	if (layer == TOP_LAYER) {
		ortho = gOrthoU[nFrame];
		sensM = gSensMU[nFrame];
		sensN = gSensNU[nFrame];
		sensP = gSensPU[nFrame];
		sensQ = gSensQU[nFrame];
		p_gSumCon = gSumConU;
	}

	if (layer == BOTTOM_LAYER) {
		ortho = gOrthoD[nFrame];
		sensM = gSensMD[nFrame];
		sensN = gSensND[nFrame];
		sensP = gSensPD[nFrame];
		sensQ = gSensQD[nFrame];
		p_gSumCon = gSumConD;
	}

	if (gAlgo == OLD_ALGO) {
		index = 0;
		for (y=0; y<16; y++) {
			for (x=0; x<16; x++) {
				ortho[index] = (sensM[y] + sensN[x]);	// simple back projection
				if (ortho[index] > 0xFF)
					ortho[index] = 0xFF;
				index++;
			}
		}	// end OLD_ALGO TECHNIQUE
		return;
	} // end if Algo = OLD_ALGO

	// LBP with Masking
	index = 0;
	gSumCon = 0;
	for (y=0; y<16; y++) {
		for (x=0; x<16; x++) {
			// MASKS OFF THE ORTHOGONAL VALUES BELOW THRESSHOLD.
			//	BOTH ORTHOGONAL SENSOR MUST ABOVE THRESSHOLD
			if ( (sensM[y] > THRESSHOLD_Y) && (sensN[x] > THRESSHOLD_X) &&
				((sensP[y-x+OFFSET_P] & sensQ[x+y+OFFSET_Q])==0x80) )		{

				ortho[index] = (sensM[y] + sensN[x]);	// simple back projection
```

```
                    if (ortho[index] > 0xFF)
                        ortho[index] = 0xFF;

                    gSumCon += ortho[index];
                }
                else
                    ortho[index] = 0;

                index++;   // point to next data
            }
        }

        p_gSumCon[nFrame] = gSumCon;

        // MASKING USING NEW TECHNIQUE
        if (gAlgo == NEW_ALGO) {
            index = 0;
            for (y=0; y<16; y++) {
                iQ = y+OFFSET_Q;          // Calculate once
                iP = y+OFFSET_P;

                for (x=0; x<16; x++) {
                    //iQ = x+y+OFFSET_Q;
                    //iP = y-x+OFFSET_P;

                    triangleCount = 0;

                    if (sensQ[iQ] & 0x80)  {
                        if (sensP[iP+1] & 0x80)           // A
                            triangleCount++;
                        if (sensP[iP] & 0x80)         // C
                            triangleCount+=4;
                        if (sensP[iP-1] & 0x80)           // E
                            triangleCount++;
                    }

                    if (sensP[iP] & 0x80) {
                        if (sensQ[iQ+1] & 0x80)           // B
                            triangleCount++;
                        if (sensQ[iQ-1] & 0x80)           // D
                            triangleCount++;
                    }

                    ortho[index] = ortho[index] * triangleCount / 8;

                    index++;
                    iQ++;               // To facilitate the +x,
                    iP--;         //                          and -x

                } // end x
            } // end y
        }// END NEW MASKING TECHNIQUE
```

APPENDIX C

Program listing to obtain Velocity Profile

```
// CCFpixel      - Cross Correlation Frequency domain

void CCFpixel(void)
{
      int p, k, y, x;
      long max;//, temp;

      long data1i[N], data2i[N], data1[N], data2[N], rr[N], ri[N];

      long tr, ti;
      long xwr, xwi;          // temporary for calculation of W * X

      int stage, group, but, n;
      int nGroup, nBut, butDistance;

      int n_nBut, but_nGroup;

      int length, count;
      int *pLag, *src1, *src2;
      int lagm[16], lagn[16];

      if (gVel == VEL_PIXEL) {
            length = 256;
            pLag = gLag;
            count = 0;
            }
      else {
            length = 16;
            src1 = &gSensMU[0][0];
            src2 = &gSensMD[0][0];
            pLag = lagm;
            count = 0;
            }

CCF_SENS:
      // CCF
      for (p=0; p<length; p++) { // REPEAT 256 - (Pixel) OR  16 x 2 - (Sensor)

            // Two FFTs for the price of one...
            if (gVel==VEL_SENSOR) {

                  for (k=0; k<N; k++) {
                  rr[k] = src1[gOrMask[k]*length+p];
                  ri[k] = src2[gOrMask[k]*length+p];

                  if (ri[k]<25)
                        ri[k] = 0;
                  else
                        ri[k] /= 8;

                  if (rr[k]<25)
                        rr[k] = 0;
                  else
                        rr[k] /= 8;
                  }
```

```
		}
		else {		// VEL_PIXEL

			for (k=0; k<N; k++) {
				rr[k] = gOrthoU[gOrMask[k]][p]/8;
				ri[k] = gOrthoD[gOrMask[k]][p]/8;
			}
		}

		nGroup = N/2;  // number of groups in each stage
		nBut = 1;		// number of butterflies in each group
					// always starts with 1 butterfly (adjacent values in sequence)

		// RFFT
		for (stage=0; stage<gLog2n; stage++) {
			butDistance = nBut * 2;		// distance between 2 butterflies

			for (group=0; group<nGroup; group++) {

				n = group * butDistance;		// starts with 0, then go on to the next butterfly
				for (but=0; but<nBut; but++) {

					n_nBut = n+nBut;
					but_nGroup = but*nGroup;

					xwr = ((rr[n_nBut] * gwr[but_nGroup]) - (ri[n_nBut] *
gwi[but_nGroup]))/1024;
					xwi = ((ri[n_nBut] * gwr[but_nGroup]) + (rr[n_nBut] *
gwi[but_nGroup]))/1024;

					// temporary save the x[n] value
					tr = rr[n];
					ti = ri[n];

					// butterfly top part
					rr[n] = tr + xwr;
					ri[n] = ti + xwi;

					// butterfly bottom part
					rr[n_nBut] = tr - xwr;
					ri[n_nBut] = ti - xwi;

					n++; // next butterfly in the group
					//count++;// each count is 1 complex multiply
							//			1 complex addition
							//			1 complex subtraction
				} // end but

			} // end group

			nBut *= 2;		// number of butterflies doubles after each stage
			nGroup /= 2;	// number of groups halves after each stage

		} // end stage

		// Extract fft results from rr, ri
		data1[0] = rr[0];
		data1[N/2] = rr[N/2];
		data1i[0] = 0;
		data1i[N/2] = 0;
```

```
data2[0] = ri[0];
data2[N/2] = ri[N/2];
data2i[0] = 0;
data2i[N/2] = 0;

for (k=1; k<N/2; k++) {
	data1[k] = (rr[k] + rr[N-k]) / 2;
	data1i[k] = (ri[k] - ri[N-k]) / 2;
	data2[k] = (ri[k] + ri[N-k]) / 2;
	data2i[k] = (rr[N-k] - rr[k]) / 2;
}

for (k=(N/2)+1; k<N; k++) {
	n = N-k;
	data1[k] = data1[n];
	data1i[k] = -data1i[n];
	data2[k] = data2[n];
	data2i[k] = -data2i[n];
}

// C
for (k=0; k<N; k++) {	// Multiply 2 sequences to cross correlate in freq domain

	//data2i[k] = -data2i[k];		// conjugate of 2nd sequence
	data1i[k] = -data1i[k];		// conjugate of 2nd sequence

	rr[k] = data1[k] * data2[k] - data1i[k] * (data2i[k]);
	ri[k] = data1i[k] * data2[k] + data1[k] * (data2i[k]);
}
// C-C : 190 us

for (k=0; k<N; k++) {
	data1[gOrMask[k]] = rr[k];
	data1i[gOrMask[k]] = ri[k];
}

nGroup = N/2;  // number of groups in each stage
nBut = 1;	// number of butterflies in each group
		// always starts with 1 butterfly (adjacent values in sequence)

// IFFT
for (stage=0; stage<gLog2n; stage++) {
	butDistance = nBut * 2;		// distance between 2 butterflies

	for (group=0; group<nGroup; group++) {

		n = group * butDistance;	// starts with 0, then go on to the next butterfly
		for (but=0; but<nBut; but++) {

			n_nBut = n+nBut;
			but_nGroup = but*nGroup;

			xwr = ((data1[n_nBut] * gwr[but_nGroup]) + (data1i[n_nBut] *
gwi[but_nGroup]))/1024;
			xwi = ((data1i[n_nBut] * gwr[but_nGroup]) - (data1[n_nBut] *
gwi[but_nGroup]))/1024;

			// temporary save the x[n] value
			tr = data1[n];
```

```
                        ti = data1i[n];


                        // butterfly top part
                        data1[n] = tr + xwr;
                        data1i[n] = ti + xwi;


                        // butterfly bottom part
                        data1[n_nBut] = tr - xwr;
                        data1i[n_nBut] = ti - xwi;

                        n++; // next butterfly in the group
                        //count++;// each count is 1 complex multiply
                                        //                      1 complex addition
                                        //                      1 complex subtraction
                    } // end but

                } // end group

                nBut *= 2;      // number of butterflies doubles after each stage
                nGroup /= 2;    // number of groups halves after each stage

        } // end stage


        max = -1;
        for (k=0;k<N;k++) {
                if (max<data1[k]) {
                        max = data1[k];
                        if (k == N || k == 0)
                                pLag[p] = 0;
                        else
                                pLag[p] = k;//+LAGOFFSET;
                }
        }


}       // end length time ccf

if (gVel == VEL_SENSOR && count==0) {
        src1 = &gSensNU[0][0];
        src2 = &gSensND[0][0];
        pLag = lagn;
        count++;
        goto CCF_SENS;
}

// Back Project to obtain Velocity Profile
if (count != 0) {
        for (y=0; y<16; y++) {
                for (x=0; x<16; x++) {
                        gLag[y*16+x] = (lagm[y]+lagn[x])/2;//+LAGOFFSET;
                }
        }
}
```

APPENDIX D

Program listing to obtain Mass Flow Rate

```
//          Prepare Co-efficient Table
//          ============================
            gDistance = 42;
            gDensity = 147;     // in 10exp(-6) // flow cylinders

            // Get maximum frame in Upstream
            gTemp = 0;
            for (i=0; i<N; i++) {
                  if (gTemp<gSumConU[i]) {
                              gTemp = gSumConU[i];
                              gMaxFrameU = i;
                        }
            }

            // Get maximum frame in Downstream
            gTemp = 0;
            for (i=0; i<N; i++) {
                  if (gTemp<gSumConD[i]) {
                              gTemp = gSumConD[i];
                              gMaxFrameD = i;
                        }
            }

            // Convert max con prof -> new algo con prof
            for (i=0; i<256; i++) {
                  gTemp = 0;
                  for (j=gMaxFrameU-1; j<gMaxFrameU+1; j++)  {
                        gTemp += gOrthoU[j][i];          // gOrthoU already new algo
                  }
                  gAvgCon[i] = gTemp / 2;
            }

            // Convert max con prof -> new algo con prof
            for (i=0; i<256; i++) {
                  gTemp = 0;
                  for (j=gMaxFrameD-1; j<gMaxFrameD+1; j++)  {
                        gTemp += gOrthoD[j][i];          // gOrthoD already new algo
                  }
                  gTemp /= 2;
                  gAvgCon[i] += gTemp;
                  gAvgCon[i] /= 2;
            }

            for (i=0; i<256; i++) {
                  gMFRprof[i] = (Uint32)gAvgCon[i] * (Uint32)gAreaRef[i];

                  if (gLag[i] != 0)// to prevent divide by zero
                        gMFRprof[i] = gMFRprof[i] / (gLag[i]);
                  else
                        gMFRprof[i] = 0;
            }
```

```
gMFR = 0;
for (i=0; i<256; i++) {
        gMFR += gMFRprof[i];
}

// conversion from sensor voltage to area profile
gMFR /= 254;            // 256 is maximum pixel value
//gMFR *= MFR_CONSTN;     // Nominator
//gMFR /= MFR_CONSTD; // Denominator

gMFR *= gDensity;
gMFR /= gAvgPeriod;
gMFR *= gDistance;

gMFR /= 1000;
```

APPENDIX E

Program listing of User Interface Unit

APPENDIX F

3rd International Symposium on process tomography paper